Section 2
Hardware Detail

Systems Unit 2.1

Contents

Illustrations

Introduction

The Systems Unit is the area of the computer which houses the majority of the components. It contains the 8086 processor, the system memory, the 3.5 inch MicroFloppy disk drive, a Power Supply Unit, the integral LCD module and all the necessary circuitry to interface to the keyboard, mouse, michrophone and all other peripheral equipment (printers, plotters, external modems, colour monitor, etc).

This chapter describes the physical and electrical details of this unit.

Details

Mechanics

All the electrical and electronic equipment is contained within the plastic case. Most of the electronics are contained on three circuit boards:

1. CPU and Display Board
2. Interface Board
3. IR Receiver Board

The CPU and Display Board and the Interface Board are mounted behind and parallel with the LCD module (i.e. slightly offset from vertical - see Figure 1).

The CPU and Display Board is mounted directly behind the LCD module. The Interface Board is mounted behind the CPU and Display Board.

Access to the CPU and Display Board is made by simply removing the Portable's front facia which also forms the LCD module housing. This is secured in place by the two screws accessible from the rear of the unit.

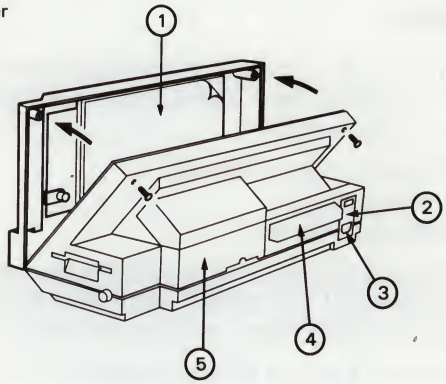
The back panel plastic assembly can also be removed once the screws are released. This is simply achieved by tilting the back panel plastic towards the rear until it is unclipped from its retaining lugs in the base plastic.

The LCD module is connected to the CPU and Display Board via a ribbon cable assembly. Disconnecting this allows free access to the back of the CPU and Display Board. The front panel LED indicators are all mounted on this side of the board.

To gain access to the component side of this board and to the remainder of the machine, a single screw, located just right of the board centre, must be removed. The CPU board then can be lifted out of its positioning slot and tilted forward.

The Interface board is secured to a metal chassis assembly by three screws.

1. LCD Module
2. Mains Switch and Fuse holder
3. Mains Input
4. Heat Sink
5. Expansion Cover



1. Interface Board
2. CPU and Display Board
3. Power Supply Unit
4. Metal Chassis Assembly
5. Disk Drive
6. IR Detector Board

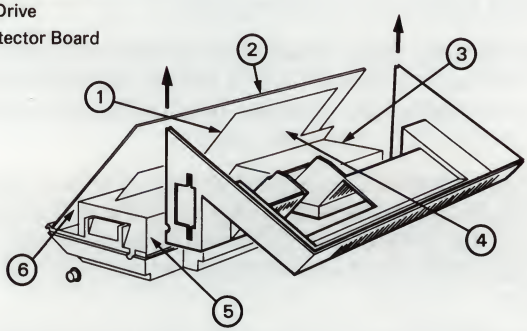


Figure 1. Systems Unit

The IR Receiver Board is a small circuit board bolted to the side of the disk drive. Photodiode detectors are mounted on the board. They project through a slot at the front of the base plastics to capture infra-red transmissions from the Keyboard or Mouse. One of the detectors is surrounded by a wide-angle lens; the other two are encased by light pipe sockets.

The Power Supply Unit (PSU), disk drive and speaker are mounted in the base of the unit. The speaker is fixed in position by metal clips. The PSU and disk drive are secured to the base plastics by screws which are accessible from underneath the unit.

There is no cooling fan; the machine relies on the natural airflow created in and around the machine by the positioning of vents in the case relative to the components which generate most of the heat. The majority of waste heat generated by the PSU is dissipated via the action of the heatsink bolted onto the Unit.

Connectors

The connector for the mains power input, and the connectors for standard peripheral units (printers, plotters, external modems, display monitors, etc) are all located on the rear panel of the Systems Unit.

Connectors for all peripheral units are located behind the Expansion cover. These are:

1. An RS232C serial communications port (25-pin female D-type).
2. A Display Unit connector for a Monitor (9-pin male D-type).
3. A Parallel Printer Port (36-way female Centronics connector).
4. The Apricot compatible Expansion Slot.

Cut-outs are provided in the Expansion cover to allow cables (for the printer, communications equipment, display monitor, etc) to be neatly and tidily routed out of the back of the unit.

The mains fuse is accesed by prising open the hinged panel that protects and surrounds both switch and socket.

CPU and Display Board

The CPU and Display Board, as the title suggests, incorporates the circuitry which performs the general processing functions within the system (CPU, system RAM, ROMs, etc) and controls the display outputs. The Expansion Slot is also located on this board.

It is linked to the Interface Board via a ribbon cable assembly. Other ribbon cable assemblies connect the board to the LCD module and to the 9-pin D-type display connector which is mounted on the metal chassis assembly.

The major circuit components on the board are:

1. The Intel 8086 processor operating at 5 MHz.
2. An Intel 8237 four channel DMA controller.
3. An Intel 8259A interrupt controller.
4. The Boot PROMs, which contain the system code for bootstrap loading and BIOS driver routines.
5. The system RAM (either 256 or 512 KBytes of dynamic RAM).
6. A DRAM controller (TMS 4500) for refreshing the system RAM array.
7. A Motorola 6845 display controller for refreshing a second display.
8. A custom gate array for driving the LCD module.
9. Either 64 Kbytes or 128 Kbytes of display DRAM, depending on the model within the Portable range.
10. A 16 x 4-bit static RAM, which is used as a programmable Colour Palette.
11. An addressable, 8-bit Latch, which is used as a Control Port for the LEDs.
12. A 15 MHz oscillator which forms the fundamental frequency source for the processor.
13. A 14 MHz oscillator which is used to drive the display circuitry.

Other ancilliary circuitry includes; data latches, multiplexers, transceivers, counters, and decoders (a mixture of PALs and also more standard logic elements). These are used as support control circuitry for synchronising signals on the System Bus, buffering transfers, general purpose timing, etc.

The chapters following provide a more detailed description of both the CPU and Display Board and the Interface Board and of their main functional components.

Interface Board

The Interface Board contains the majority of peripheral and interface circuitry for the system. The board is linked to the CPU and Display Board via a ribbon cable assembly.

The major circuit components on the board are as follows:

1. Western Digital WD2797-02 Floppy Disk controller.
2. Zilog Z80 SIO/0 dual channel serial communications device, which forms the interface for RS232C communications and the input data supplied from the IR Detector Board.
3. An Hitachi 6301 single chip microcomputer, which forms the interface between the voice input circuitry (microphone, analogue-to-digital converters, etc) and the software speech processing system.
4. A Texas Instruments SN76489 sound generator which is linked via an audio amplifier and two-wire cable assembly to the internal speaker.
5. A parallel printer port implemented by an 8-bit data latch, octal buffer and a data selector.
6. An Intel 8253A-5 programmable timer which is used to set the baud rates for the RS232C port, the sampling rate of the a-d converter in the voice input circuitry, and the regular system clock interrupt (20 ms cycle).

Other ancilliary circuitry includes; data latches, data buffers, data selectors, decoders and various logic gates. These are used as support control circuitry for the interfaces described above.

Infra-Red Detector Board

The Infra-Red Detector Board is bolted onto the disk drive. It is linked to the Interface Board by a 4-wire cable assembly. This provides +12V and +5V dc regulated supplies to the board and also carries the decoded IR signal pulses to the Interface Board.

The Board incorporates three photodiode detectors, an amplifier section, and a timer circuit (see Circuit Diagram in Appendix D). It converts input infra-red pulses into a form suitable for use by the receiver circuits on the Interface Board.

Two of the photodiodes are fitted behind sockets. These sockets are for connecting light pipes to the Systems Unit.

A third photodiode detector is surrounded by a wide-angle lens in order to optically amplify freespace infra-red transmissions.

Fitting a light-pipe into the right hand photo-diode socket operates a switch which switches the diode surrounded by the lens off. This action is necessary to reduce the chance of interference from other infra-red sources in multiple machine environments.

Detected infra-red pulses are converted into electrical pulses by the diodes. To account for the variations in signal strength of the input infra-red pulses, and to prevent saturation of the diodes, the current through the diodes is regulated by an a.g.c amplifier (Q1).

Following conversion to a voltage, the raw input pulses are amplified by a high gain amplifier circuit (Q2, Q3). This produces a pulse output with an amplitude of approximately 2 to 5 V which is then supplied to a 555 timer circuit.

The timer is wired in a non-retriggerable mode to prevent false triggering once a pulse edge is detected. The timer "squares" up the raw input pulses of 18 to 20 μs duration and converts them into 25 μs duration output pulses. These are supplied to the Interface Board.

On the Interface Board, the transmitted signal is separated into timing and data pulses, so that the data signal can be clocked into the Z80 SIO as a standard monosync transmission.

Disk Drive Unit

The MicroFloppy Disk Drive Unit is mounted onto the base plastics of the Portable. The disk eject button of the Drive Unit fits through the side panel to provide the means for ejecting disks out of the disk drive.

A ribbon cable assembly connects the disk drive to the disk interface on the Interface Board. A second, 4-way cable assembly supplies power from the PSU to the Drive Unit.

The MicroFloppy Disk Drives use 80-track double sided 3.5 inch MicroFloppy disks with a total storage capacity of 720 Kbytes of formatted data. The BIOS is also configured to allow the user to read, write and format 70-track single sided 3.5 inch MicroFloppies.

The DISC indicator on the front panel of the Portable is under direct software control of the BIOS. The indicator is switched on every time a signal is sent to load the disk drive heads and remains so until the drive heads are unloaded.

Expansion

An Expansion Slot is provided for extending the System Bus (address, control, and data lines). It is designed to take any of the current ACT Expansion Boards.

The Expansion Slot is a 64-way connector (DIN 41612, 2 by 32-way female with a type B housing). The signal lines connected to the Expansion Slot are 95% functionally compatible with all other machines in the current range of Apricot microcomputers. (The main area of difference is that there are no external DMA facilities available on the Portable).

Power Supply

General

The mains power input is connected to the Systems Unit via the 3-pin male IEC mains inlet connector on the rear panel. The mains input is fed through a line filter via the input fuse and mains switch before being supplied to the rectifying and regulating circuitry of the Power Supply Unit (PSU).

The PSU rectifies mains supplies of either 110 Volts or 240 Volts AC. The unit is reconfigured to the appropriate mains voltage by a switch selection on the base of the Portable.

The PSU is rated at 35W, is of switched mode design, and provides regulated outputs of +12V, +5V and —12V for use by all the boards and components within the Systems Unit. This includes the CPU and Display Board, the Interface Board, Disk Drive Unit, the Infra-Red Detector Board, and any expansion boards that are fitted to the Expansion Slot.

All other external devices connected to the Portable, are normally supplied with mains directly to their appropriate input.

All power supply components are housed in a shielded case. A fuse is located in the mains input line, within the PSU. This is in addition to the user accessible mains input fuse within the mains switch housing.

DC Supply Distribution

The regulated outputs from the PSU are supplied to the System Board via three cable assemblies. Both ends of each cable assembly are terminated by Molex connectors. The cable assemblies route the dc supplies to the:

1. CPU and Display Board
2. Interface Board.
3. Disk Drive.

The Interface Board supplies dc to the IR Detector Board via the board wiring and another cable assembly terminated with Molex connectors.

The PSU provides regulated supplies on each of the Molex connectors of +5V dc, +12V dc, and —12V dc. The maximum current ratings for the various supplies are detailed below.

1. +5V supply 4A
2. +12V supply 1.0A (1.5A peak)
3. —12V supply 0.25A

Fuse rating

The rating of the fuse on the rear panel of the Systems Unit is as detailed below.

240V mains input - 2A, 5 x 20 mm fast action.
115V mains input - 2A, 5 x 20 mm fast action.

Dimensions

Height:	7.9 inches	(200 mm)
Width:	17.7 inches	(450 mm)
Depth:	6.8 inches	(172 mm)
Weight:	10.0 lbs	(4.54 kg)

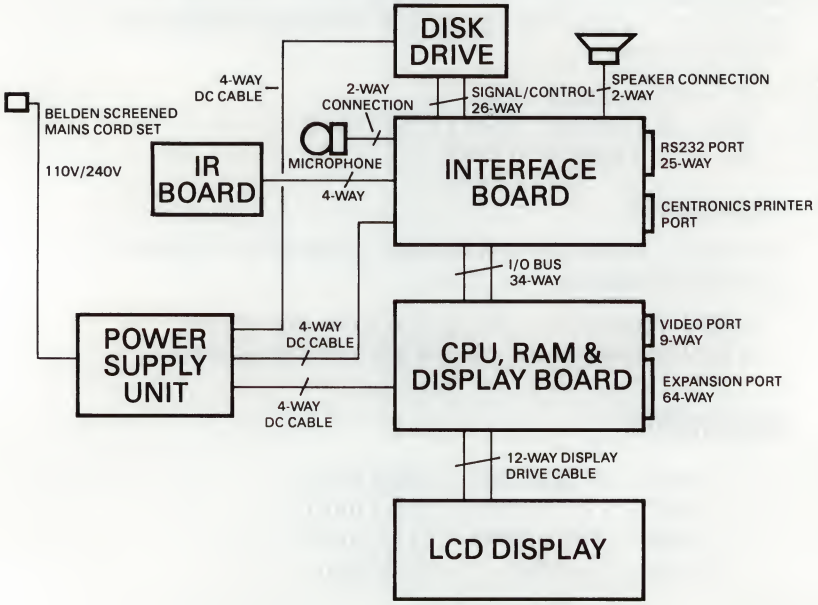


Figure 2. Systems Unit Schematic

System Details 2.2

Contents

Illustrations

Introduction

The purpose of this chapter is to present an overview of the principal circuit elements on the two major circuit boards in the Portable; the CPU and Display Board and the Interface Board. The programmable elements are then broken down into further detail in subsequent chapters. This chapter also describes various miscellaneous areas of circuitry which do not warrant a chapter of their own.

Other details included in this chapter are a full list of memory and I/O port addresses as allocated within the Apricot Portable.

Details

General

The majority of the circuitry within the Portable is split between two boards; the CPU and Display Board and the Interface Board.

The CPU and Display Board contains the processing elements, the Expansion Slot, the system RAM, Boot ROMs and all the necessary circuitry for driving the displays.

The Interface Board contains all the peripheral interfaces for; keyboard/mouse data, RS232 communications, printers, sound, speech input, and Floppy disk control.

The devices on the two main circuit boards, and the control, timing and interface circuitry associated with them, are interconnected with the Intel 8086 processor using the standard three parts of a 16-bit bus-based architecture; a 16-bit Data Bus; a 20-bit Address Bus; and a multi-purpose Control Bus.

Figure 1 represents this architecture in a block schematic diagram. It shows all the major peripheral areas of circuitry which are linked to the CPU. For simplicity, the diagram omits the detail of buffering and timing and control signals.

The processing system operated on the board is a real time interrupt driven system primarily based on the interrupt structure of the 8086 and the interrupt facilities provided by an Intel 8259 interrupt controller.

Peripheral support for the 8086 is provided by a mixture of intelligent support chips and combinations of simpler standard logic elements.

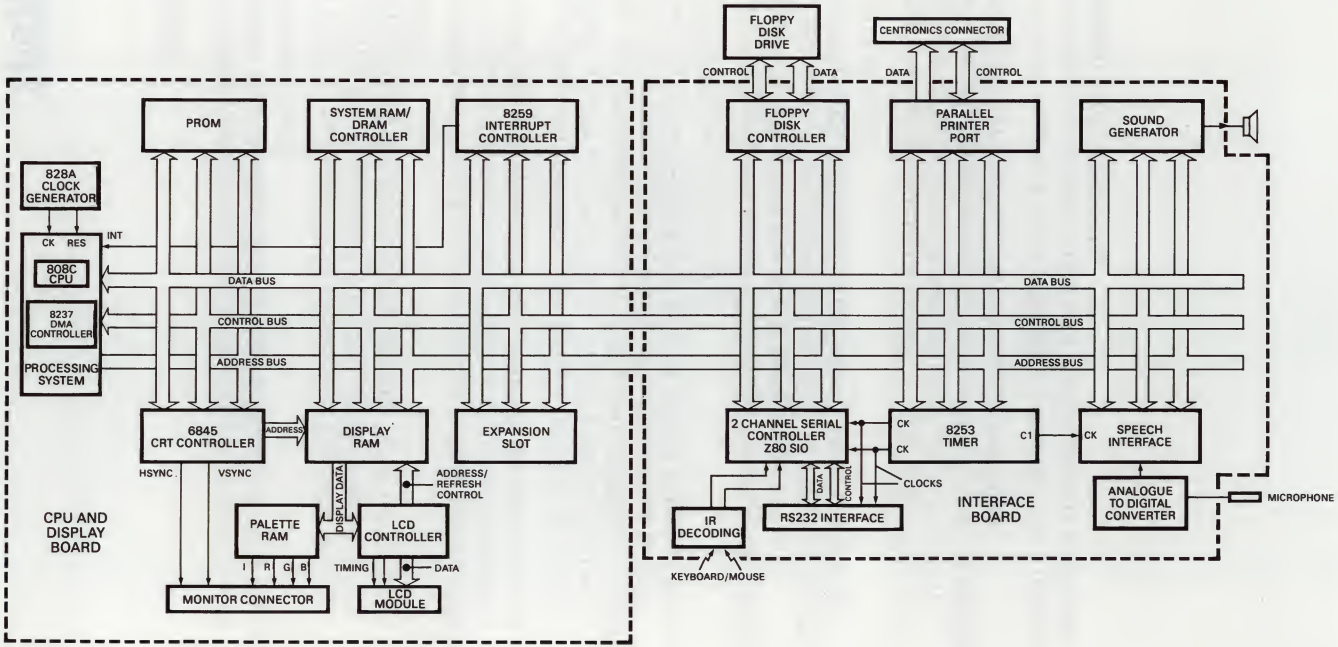


Figure 1. System Detail: Data Flow Schematic

Located on the CPU and Display Board are the following components:

1. An Intel 8086 processor running at 5 MHz.
2. An Intel 8259 Interrupt Controller.
3. An Intel 8237 4-channel DMA Controller.
4. An Intel 8284 clock generator/driver.
5. 256 Kbytes of system DRAM (dynamic RAM) — 512 Kbytes in the second model of the range.
6. A DRAM controller (TMS4500).
7. 32 Kbytes of ROM which store the code for the ROM based BIOS (expandable to 64 Kbytes).
8. Either 64 or 128 Kbytes of DRAM which functions as display RAM.
9. A 16 x 4-bit static RAM which is integrated into the display control circuitry for selecting the colours on the optional display monitor. This is termed the palette.
10. A Motorola 6845 for controlling the optional display monitor and generating the memory addresses for accessing the monitor display data.
11. A custom gate array which performs the role of display controller for the LCD, controller for access to the Display RAM and Display DRAM array refresh generator.

Other ancilliary circuitry on the CPU and Display Board include; PALs for control and address decoding, an I/O mapped latch which is used to drive the LEDs, data latches, multiplexers, transceivers, decoders, plus other pieces of "glue" logic. These devices are used in the supporting control circuitry for synchronising signals on the System Bus, and buffering transfers to the Interface Board/Expansion Slot.

The peripheral circuitry on the Interface Board includes:

1. A Western Digital WD2797-02 Floppy Disk Controller for controlling the MicroFloppy Disk Drive.
2. A Zilog Z80 SIO/0 dual channel serial controller, which interfaces to the RS232C port, receives keyboard/mouse data, and also provides an extension to the interrupt structure.
3. An Intel 8253 Timer which provides the baud rate clocks for the RS232C port, sets the sampling rate for the Speech Interface circuit, and also produces the regular system clock interrupt.
4. Speech Interface circuitry, which is made up of a mixture of analogue and digital circuitry. It consists of audio amplifiers, analogue-to-digital converters and an Hitachi 6301 single-chip microcomputer.
5. A Texas Instruments SN76489 multi-channel sound generator.
6. A parallel printer port constructed from discrete 74LS components.

Other areas of circuitry on the Interface Board include a bit addressable latch which acts as a multi-function control port; data latches, and transceivers; and decoders plus other pieces of "glue" logic. These devices are used in the supporting control circuitry for synchronising signals on the System Bus, and buffering transfers to external peripherals.

Processor

Details of the Intel 8086 processor are widely available. It is a true 16-bit processor, directly language compatible with the more commonly used 8088. It possesses:

1. 16-bit wide internal register architecture.
2. 16-bit wide external data bus.
3. Segmented addressing structure to support modular programming.
4. The capability of addressing 1 Mbyte of memory space and 64 Kbyte of system I/O.

The 8086 is configured to operate in minimum mode (i.e. it does not support a multi-processing configuration). It's basic clock frequency is 5 Mhz, which is derived from a 15 Mhz oscillator and supplied from an Intel 8284 clock generator.

DMA controller

The DMA controller is an Intel 8237 4-channel device. Supporting circuitry includes a decoder and latch for stretching the range of the controller across its' basic 64K boundary. Its' main function is for transferring disk data between memory and the Floppy Disk Controller, but it can also be programmed to perform memory-to-memory transfers as required.

Memory

The base model of the Portable range is fitted with a minimum of 256 Kbyte of system RAM (using 64K x 1-bit DRAMs). This is expandable by fitting one of the standard Apricot RAM expansion boards into the Expansion Slot.

The second model in the Portable range is fitted with 512 Kbyte RAM as standard using 256K x 1-bit DRAMs. This is expandable to 768K using an Apricot 256K RAM Expansion Board.

In the 256 Kbyte model of the Portable range, the System RAM is formed by two banks each of 128 Kbyte; one mapped into the address range 00000H to 1FFFFH, the second into the address range 20000H to 3FFFFH. Each bank is composed of sixteen 64K x 1-bit DRAMs (one bit per data line).

In the 512 Kbyte model of the F1 range, the System RAM is formed by a single bank of 256K x 1-bit DRAMs (half the board is left unpopulated). In this model The System RAM occupies the address range 00000H to 7FFFFH.

The system RAM is located at the bottom end of the available memory address range and is sub-divided into a number of allocated memory areas. These are specified by the hardware/BIOS as follows:

Address	(hex)	Use
00000 —	003FF	Interrupt vectors
00400 —	027FF	BIOS working area (pointers/ASCII screen images)
02800 —	05FFF	BIOS data and stack
06000 —	3FFFF *	DOS/application/user area

* 06000 — 7FFFF on 512 Kbyte Portable, both models expandable to CFFFFH.

Not all of the available System RAM is available to the user or application.

The lowest 1 Kbytes are reserved for the standard Intel interrupt vectors (4 bytes per entry).

All other defined areas are allocated by the ROM based BIOS/operating system interface (see the "Guide to the BIOS" chapter for further details).

Processor accesses to the system RAM are interleaved with DRAM array refreshing and are controlled by a proprietary DRAM controller (a TMS 4500). The controller generates all the necessary timing signals (RAS, CAS, etc), and address lines for accessing the DRAMs and refreshing the DRAM array.

The System RAM area is not the only area of memory within the Portable. Three other areas exist. One area is the Display RAM, the second is the palette RAM, the third is the Boot ROMs.

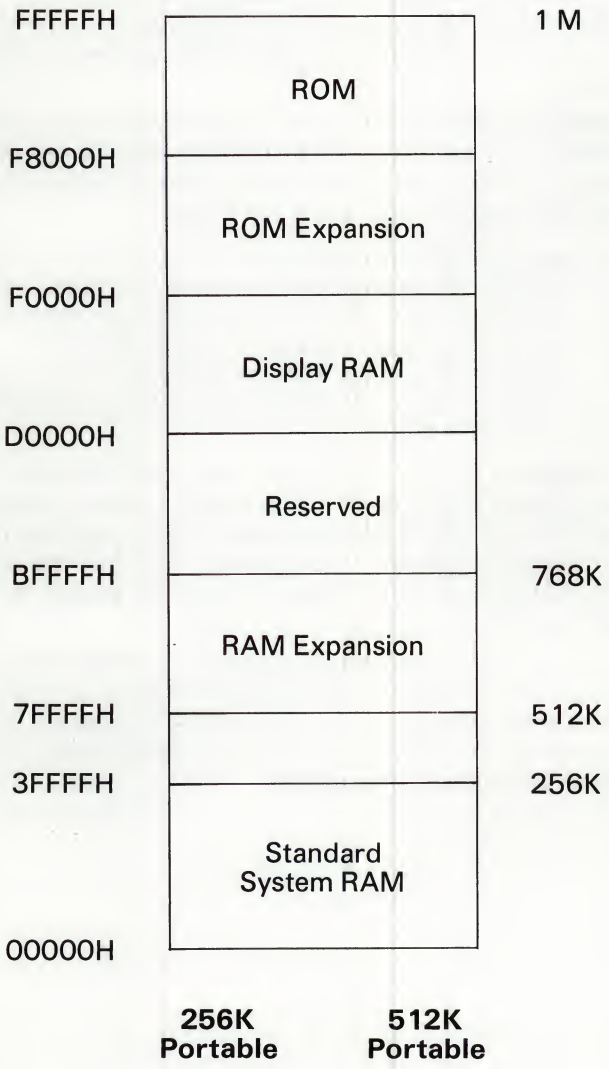
The Display RAM in the base model of the range is formed by 64K DRAM starting at address location E0000H and stretching up to EFFFFH. In the second model, the display RAM is formed by 128K DRAM starting at D0000H and stretching up to EFFFFH. A description of the usage of the display RAM is provided in the following section on Display Control.

The palette is a small area of I/O mapped RAM formed by a 16 x 4-bit static RAM. It is used to determine the colour mix at the colour display outputs. It occupies a single port address in the available I/O address space, at 2AH. (The RAM data lines are wired to the 4 LSB data lines of the system data bus, the address lines are wired to the MSB data lines of the low order section of the bus).

The Boot ROMs are located at the top of the available memory addresss space. The Boot ROMs contain the bootstrap loader, diagnostics, calculator software and the BIOS code/device driver routines for handling the standard hardware.

The BIOS code is currently contained in two 16K x 8-bit ROMs. The board is tracked to take 32K x 8-bit ROMs to allow for future BIOS expansion.

Memory Map


FFFFFH
1 M
ROM
F8000H
ROM Expansion
F0000H
Display RAM
D0000H
Reserved
BFFFFH
768K
RAM Expansion
7FFFFH
512K
3FFFFH
256K
Standard
System RAM
00000H
256K
Portable
512K
Portable

Wait States

This is controlled via a shift register and the Intel 8284 which drives the ready lines of the CPU and DMA controller. A minimum of one wait state is automatically inserted into all memory and I/O transfer operations to and from the processor/DMA Controller. This is extended in some cases to four wait states to suit the speed of slower peripherals on the boards.

The DRAM refresh cycle for the system RAM by the DRAM controller is transparent to CPU/DMA controller operations. Therefore, all CPU/DMA accesses to read or write from the System RAM always incurs one wait state.

CPU accesses to read from and write to locations in the Display RAM are interleaved with accesses from two refresh circuits, which take precedence over processor accesses. One of the refresh circuits is used to refresh the Colour Display; the second to access the memory to update the LCD. This LCD access cycle also automatically performs the refresh for the DRAM array.

As processor accesses are asynchronous with the two refresh cycles, the number of wait states inserted when accessing the display RAM is variable, being dependent on whether a refresh cycle is in progress or not. The minimum number of wait states inserted on a memory access is one; the maximum is 3.

External devices connected to the Expansion Bus can extend the basic one wait state to suit the slower speed of it's own peripherals, by activating the appropriate Expansion Bus input ready line (IORDY for I/O mapped peripherals; MRDY for memory mapped peripherals).

Interrupt Control

The interrupt structure operated within the system is provided by the interrupt handling facilities within the Intel 8259 Interrupt Controller (PIC), the 8086 CPU and the Z80 SIO.

The PIC forms the interface between the devices capable of generating interrupt requests and the interrupt control line of the 8086 processor.

Interrupt requests from the various sources are supplied to the PIC, when the associated device requires a service routine to be undertaken. The sources are as follows:

1. Timer (PIT OUT0)
2. Disk Controller (FDC INT)
3. Speech Controller (6301 P22)
4. Expansion 1 (EXP 1)
5. Serial Interface (Z80 SIO INT)
6. Expansion 2 (EXP 2)
7. Parallel Printer Interface (Busy)
8. DMA controller (DMA EOP)

The PIC functions as the manager of the interrupt driven processing system; generating an active interrupt to the 8086 processor only when an incoming interrupt request has a higher priority than any interrupt service routine already in operation.

On acknowledgement from the CPU that it is able to accept the generated interrupt, the PIC supplies vectoring data to the CPU which acts as a pointer to the required service routine. The vectoring data, the mode of operation of the PIC and the priority of the interrupt requests are set up by the BIOS.

The CPU responds to interrupt requests by implementing a two-stage interrupt acknowledge cycle, which causes the PIC to supply the appropriate interrupt vector onto the System Data Bus.

The various interrupts can be enabled and disabled individually by program commands to the PIC. All these interrupt sources can be masked as a group by clearing the Interrupt Enable Flag in the CPU. This action disables the Interrupt Request (INTR) input to the CPU.

The Z80 SIO acts as an extension to this interrupt process and handles interrupt requests associated with:

1. Serial communications via the RS232C interface.
2. The keyboard/mouse data input channel

Once the BIOS has detected that the source of the interrupt is from the Z80 SIO, it can then read a register within the SIO itself to detemine the actual process requiring servicing.

Display Control

General

The display architecture of the Apricot Portable is totally graphics oriented (i.e. based on a bit-map instead of characters) and can be software configured for a number of different resolutions and modes.

In both models in the Portable range, the programmer has the facility to drive both the flat panel display (LCD module) and an optional colour monitor with different data and produce two different displays simultaneously.

The Portable does not use the more commonly found method of dual-porting the System RAM for display purposes (as found on the Apricot pc/xi range and the Apricot F1). Instead it utilises a separate area of memory, designated as display RAM.

The base model of the Portable range is fitted with 64K of display memory. This allows the programmer to use the colour modes, resolutions and display features as detailed in the next two paragraphs. The resolutions described match the resolutions of the current ACT colour monitors.

For USA usage and other countries using 60 Hz mains supply frequency:

1. 640 x 200 bit-mapped colour graphics using up to 2 colours + 640 x 200 bit-mapped LCD.
2. 640 x 200 bit-mapped colour graphics using up to 4 colours, LCD off.

For UK, European and other countries using 50 Hz mains supply frequency:

1. 640 x 256 bit-mapped colour graphics using up to 2 colours + 640 x 200 bit-mapped LCD.
2. 640 x 256 bit-mapped colour graphics using up to 4 colours, LCD off.

The second model within the range incorporates the colour option circuitry, which is obtained by fitting an extra 64 Kbytes of DRAM to the Portables' internal circuitry. This is implemented as a factory fitted option.

The extra memory provides applications software with sufficient memory capacity to drive the flat panel display and an optional colour monitor using a greater number of colours than the base model.

The colour modes, resolutions and display features available to the programmer when the extra display RAM is fitted are detailed in the next two paragraphs.

For USA usage and other countries using 60 Hz mains supply frequency:

1. 640 x 200 bit-mapped colour graphics using up to 8 colours + 640 x 200 bit-mapped LCD.
2. 640 x 200 bit-mapped colour graphics using up to 16 colours, LCD off.

For UK, European and other countries using 50 Hz mains supply frequency:

1. 640 x 256 bit-mapped colour graphics using up to 8 colours + 640 x 200 bit-mapped LCD.
2. 640 x 256 bit-mapped colour graphics using up to 16 colours, LCD off.

The 128 Kbyte of available display memory space is mapped above 832K in the 1 Mbyte address range of the 8086 processor.

Bit-map Structure

The LCD and colour monitor are bit-mapped in two entirely different ways to match the required functionality of the two different display technologies. LCD data is stored as a linear sequence of bytes on even addressed byte boundaries.

The colour monitor data is stored in a much more complex arrangement due to the inherent nature of the colour image as compared with the monochrome LCD image. This is based on a segmented structure within a series of memory planes.

Memory Planes

A memory plane is a block of memory in the display RAM, and in our case is not contiguous in memory. Utilisation of the memory planes is entirely dependent on the model in the range and the mode selected. In the base model, two 32K memory planes are available. In the full colour model, four 32K planes are available.

One plane has to be assigned to the LCD to display data on the LCD. Otherwise it can be used as a colour plane, if the LCD is switched off.

Usage of the planes is entirely dependent on how many colours the programmer wishes to display on the monitor. 16 colours requires 4 planes, 8 colours requires 3 planes, 4 colours requires 2 planes, 2 colours (monochrome) requires a single plane. All other intermediate numbers of colours require the higher number of planes (e.g. 9 colours requires 4 planes).

Palette

The colour palette selects the colour presented on the colour monitor as set by the logic state on the four colour coded display outputs (standard IRGB).

The programmer masks out the planes not in use (or the planes allocated to the LCD) and also selects the colours for display by programming the palette. The function of the palette is to translate the information from the planes into the colour coded IRGB outputs.

Display Circuitry

Other areas of circuitry besides the two areas of RAM (display and palette) include a custom LCD controller and a Motorola 6845 CRT controller. These two controllers share and control access to the memory planes. CPU accesses to the display RAM have to be interleaved between the two controller access cycles.

The LCD controller is used for:

1. Driving the LCD module.
2. Accessing the LCD display data.
3. Refreshing the DRAM array (this is provided by virtue of the regular access cycle for LCD data).
4. Arbitrating between CPU, LCD and CRT controller access to the display RAM.

The CRT controller is used for:

1. Accessing the data written into the display RAM for updating the image on the colour monitor.
2. Generation of timing signals which control the display scanning of the monitor.
3. Hardware display scrolling.

Floppy Disk Control

The Floppy Disk interface on the Interface Board provides all the control functions necessary for formatting and transferring data to and from MicroFloppy Disks in the Disk Drive.

The configuration can operate with either single-sided or double-sided disks.

The Floppy Disk Interface consists of a Western Digital WD2797-02 Floppy Disk Controller (FDC), a series of buffers and control ports, and the interface connector.

Four control lines for the disk drives are generated directly under software control, via the control port on the Interface Board.

The functions for, controlling the movement of the read/write head, transferring data to and from the disks, and monitoring status signals from the drives are implemented by the FDC.

All disk transfer operations performed by the FDC (formatting, reading disk data, writing data onto disks), are controlled by the system software, and involve both the 8086 CPU and the DMA controller. The DMA controller is used to perform data transfers between system memory and the FDC on a byte-by-byte basis.

Expansion

The Portable Expansion Slot is located on the CPU and Display Board. It provides an extension of the processing system for use by optional Expansion boards. Access to the slot is made by removing the Expansion cover on the back of the Systems Unit.

A high degree of compatibilty has been maintained with the other products within the Apricot pc/xi range of computers. This is such that all existing ACT Expansion boards (Winchester Controller, Modem, RAM cards, etc) can be used within the Portable without any modification to the Expansion Board hardware.

The extension connections wired to the Expansion Slot are:

1. The 16-bit system data bus.
2. The 20-bit system address bus.
3. Various control and timing signals.
4. Power supply outputs.

The same physical connector is used for the Expansion Slot as used on all current machines within the Apricot pc/xi product range. This is a 64-way connector (DIN 41612, 2 by 32 female, with a type B housing).

Of the 64 connections routed to the slot, 60 are as on the pc/xi range of micros and are compatible connections. There are minor differences in detail. For example, Interrupt 2 on the pc/xi range is Interrupt 3 on the Portable.

The main area of difference is that DMA facilities are not routed to the Portable Expansion Connector as provided on the Apricot pc/xi range, these pins are left open circuit.

Keyboard/Mouse Data

The receive channel of channel B of the Z80 SIO is used for keyboard/mouse data input. It is programmed to operate in synchronous mode (Monosync) at a data rate determined by the incoming data stream.

It is supplied with keyboard/mouse data via the IR receiver board which decodes the incoming data and converts it into an acceptable serial waveform. (Details of the IR Receiver Board are discussed in the chapter headed Systems Unit).

A signal conditioning circuit then separates the serial waveform into data and clock signals. The data is supplied to the receive data input of channel B (RxD) and the clock signal to the receive clock input (RxCKB) to clock each data bit into the Z80 SIO.

The Z80 SIO converts the serial data into parallel format. It then generates an interrupt and produces an associated interrupt vector to signify keyboard/mouse data available.

The Keyboard formats a valid key closure into a serial packet consisting of a four byte sequence. The format is the synchronous transmission mode Monosync and is operated at a fixed data rate of approximately 3.85 Kbits/sec.

The four byte sequence consists of the Sync header byte (5AH), a status byte and two data bytes. The status byte and keycode data bytes are encoded with a Hamming format.

Using Monosync means that the Z80 SIO has to first detect a valid data pattern, (the sync header byte) before it regards the data sent as being valid. This provides a high degree of protection from other infra-red sources, as they will not contain the sync header and will therefore be totally disregarded.

Using a Hamming format to encode the data enables the BIOS software to check the integrity of the data received from the Keyboard. It produces a highly reliable system for proving the validity of the Keyboard data, providing a measure of protection against a transmission which contains a valid sync byte, but invalid data (missing or corrupted data).

The Mouse transmits data in a similar four byte sequence to the Keyboard. It uses the same sync header byte, but the data bytes provide information on mouse movement and the state of the mouse switches. The Z80 SIO is unable to differentiate between mouse and keyboard data. The BIOS software determines this by reading a flag bit in the synchronous data packet.

System reset

The system is reset by one of two methods; by powering the Portable off and on or by the System reset button on the Keyboard. Both methods produce the same effect on the circuitry. The reset control lines to the CPU and all peripheral circuitry are held active, setting all programmable elements to initialised status. When the reset line returns to it's inactive state, the CPU accesses the instruction stored at absolute address location FFFF0H to initiate the normal system start-up sequence.

The system reset button is located on the top edge of the Keyboard Unit. Holding the button down for approximately one second actions the reset circuitry of the Intel 8284 clock generator on the CPU and Display Board.

The actual "data" transmitted by the Keyboard to reset the system is a contiguous sequence of sync header bytes at a rate of one sync byte every 15.6 ms. These sync bytes are processed in the normal way of Keyboard data via the Infra-Red Detector Board, and clocked into Channel B of the Z80 SIO.

Each sync byte received by the Z80 SIO causes a pulse (logic low) on the Z80 SYNCB modem control output. In a normal data transmission, the sync byte is immediately followed by data and it's associated clocks which terminates the pulse on the SYNCB output.

When a string of sync bytes are sent with no data interleaved, the pulse on the SYNCB output is extended. (This effect is produced by feeding the SYNCB output back to the signal conditioning circuit which separates out the keyboard/mouse data from their associated clocks. Because no clock pulses are received following the sync byte, the SYNCB output alters the rate at which clock pulses are supplied to the SIO from the free-running oscillator in the conditioning circuit).

The extended SYNCB pulses gradually discharge a capacitor (C10) in the reset timing circuit on the Interface Board. After a significant number of pulses, the capacitor is discharged sufficiently to set the output of the reset timing circuit (an LS74 latch). This action generates the system reset signal, via the Intel 8284 clock generator on the CPU and Display Board.

RS232 Communications

The second channel (channel A) of the Z80 SIO is available for communication between the Portable and external equipment via an RS232C link.

The RS232C channel can be programmed to operate in either asynchronous or synchronous modes, with transmit and receive baud rates determined either via the Intel 8253 Timer or via the external data communications equipment.

The RS232C interface is able to support:

1. Asynchronous communications with 5, 6, 7 or 8 bits per character and various choices of stop bits and parity sense.
2. Bit oriented synchronous communications, such as SDLC and HDLC.
3. Byte oriented synchronous communications, such as Monosync and Bisync.

The channel can also be programmed to generate an interrupt and also provide an associated interrupt vector for a variety of conditions occuring in the RS232 channel. These include:

1. Receive characters available.
2. External/status events (e.g. modem control line changes).
3. Transmit buffer empty.
4. Various error conditions (e.g. receiver overrun, parity errors, etc).

Parallel Printer Port

The Parallel Printer Port forms the interface for parallel printers and plotters and is located on the Interface Board. The circuitry consists of a number of 74LS series logic elements and a Centronics connector.

The pin connections to the Centronics port support the standard Centronics 8-bit data interface and five of the more common handshake signals required/supplied by parallel printers. These include:

1. Data Strobe
2. Busy
3. Printer Select
4. Fault
5. Paper Empty

Speech Interface

The Speech Interface circuitry forms the interface between the user (via a microphone) and the speech driver software. It is based upon a analogue-to-digital conversion stage, controlled by a 6301 processor.

The prime purpose of the interface is to convert the analogue speech input from the microphone into digital format for use by the speech driver software.

The circuitry consists of a multi-stage signal processing circuit and a a-d converter which is based upon a sample and hold arrangement. The sample rate is controlled by a clock signal supplied from the on board Timer circuit.

The value programmed into the timer is set by the speech driver software.

The digitised audio samples required by the speech driver are read into the 6301 and transferred to the 8086 processing environment using an elaborate communication mechanism. This involves a data transceiver and the use of interrupts and handshake control signals.

System Timer

This is formed by the Intel 8253-5 Programmable Interval Timer (PIT) and is used to generate a variety of timing signals. These include:

1. A clock output (OUT0), which is connected to an interrupt request line (IR0) of the Interrupt Controller on the CPU and Display Board. The output is used by the BIOS as a general purpose system timer (20 ms).
2. Two squarewave clock outputs (OUT1 and OUT2) which can be used to set the baud rates for the RS232C serial interface. The frequencies of the two clock outputs are under direct control of the programmer.

The squarewave output from the Timer (OUT1) is used by both the RS232C interface (for implementing a split rate baud clock) and the speech interface circuitry. If split rate baud rate clocks are required, the speech interface cannot be successfully utilised. This also applies to the opposite situation.

The timer is organized internally as three independent programmable 16-bit counters. Each counter can be set for various operating modes and rates by programming data into a control register.

Sound Generation

The Sound Generator can be programmed to produce audio frequency tones, audio noise, or complex synthesized sounds in a frequency range from 500 Hz to 5 kHz. It is utilised by the BIOS to generate:

1. The keyboard keyclick.
2. A bell tone to act as a general warning signal by a number of applications.

The basis of the sound generator circuitry is formed by the Texas Instruments SN 76489 Programmable Sound Generator (SG). Internally the device features:

1. Three programmable tone generators, each with associated control registers.
2. A single programmable noise generator with associated control registers.

The outputs of the four generators are summed together internally within the chip to produce a single output line for driving the internal loudspeaker.

I/O Port Map

Port Address	Function	Access
00H	FDC status register	r/o
00H	FDC command register	w/o
02H	FDC track register	r/w
04H	FDC sector register	r/w
06H	FDC data register	r/w
08H	PIT counter 0 — system clock	r/w
0AH	PIT counter 1 — Voice clock/RS232 baud	r/w
0CH	PIT counter 2 — RS232 baud	r/w
0EH	PIT command register	r/o
10H	6301 Voice data port	r/w
12H	6301 Voice interrupt clear	w/o
14H	6301 Voice generate IRQ	w/o
16H	6301 Voice clear LP23	w/o
18H	SIO channel A data — RS232	r/w
1AH	SIO channel A command — RS232	w/o
1AH	SIO channel A status — RS232	r/o
1CH	SIO channel B data — keyboard	r/w
1EH	SIO channel B command — keyboard	w/o
1EH	SIO channel B status — keyboard	r/o
20H	Centronics data	w/o
22H	Centronics busy interrupt reset	w/o
24H	Centronics status (See detail on this port below)	r/o
26H	Sound generator	w/o
28H	LCD voltage (contrast)	w/o
2AH	Colour Palette port	w/o
2EH	CRTC/LCD/Memory control (low byte)	w/o
2EH	LED control (high byte) (See detail on this word port below)	w/o
30H	Floppy Drive select (1=drive A:, 2=B:)	w/o
32H	Floppy Head Load (active high)	w/o
38H	Floppy Drive Disk Change Reset active high pulse for at least 300μs	w/o
34H	SIO M1 signal (for int vector)	w/o
36H	6301 Voice interrupt generate	w/o
3AH	Centronics strobe output	w/o
3CH	SIO clock source select 0	w/o
3EH	SIO clock source select 1	w/o

The following group of ports are addressed as Word Registers with the high byte being the required data.

Port Address	Function	Access
40H	Channel 0 DMA address/base	r/w
42H	Channel 0 DMA word count	r/w
44H	Channel 1 DMA address/base	r/w
46H	Channel 1 DMA word count	r/w
48H	Channel 2 DMA address/base	r/w
4AH	Channel 2 DMA word count	r/w
4CH	Channel 3 DMA address/base	r/w
4EH	Channel 3 DMA word count	r/w
50H	DMA Read Status	r/o
50H	DMA Write Command	w/o
52H	DMA Write Request	w/o
54H	DMA Write Single Mask	w/o
56H	DMA Write Mode	w/o
58H	DMA Clear Byte Pointer Flip-flop	w/o
5AH	DMA Read Temporary	r/o
5AH	DMA Master Clear	w/o
5EH	DMA Write All Mask Register Bits	w/o
58H	DMA interrupt reset	r/o
66H	DMA channel 0 extension register	w/o
64H	DMA channel 1 extension register	w/o
62H	DMA channel 2 extension register	w/o
60H	DMA channel 3 extension register	w/o
68H	PIC Register 0	r/w
6AH	PIC Register 1	r/w
6AH	PIC Interrupt Mask	r/w
6CH	6845 CRTC register address port	w/o
6EH	6845 CRTC register data port	r/w

Port 02EH — CRTC Control and LEDS Control

This is a Word Write only port. A copy of the port is maintaned by the BIOS at the memory location given by adding an offset of 08H from the base address provided by the DWORD pointer located at 722H.

Bit 0 — CRTC reset

0 = Reset off
1 = Reset on

Bit 1 — LCD Select 1

0 = LCD mapped at 0X0000h
1 = LCD mapped at 0X8000h

Bit 2 — Colour Display Control

0 = Colour off
1 = Colour on

Bit 3 — LCD Display Control

0 = LCD off
1 = LCD on

Bit 4 — LCD Select 2

0 = LCD mapped at 0EX000H
1 = LCD mapped at 0DX000H

Bit 5 — Colour plane Low access control

0 = 0E0000H — 0EFFFFH Low byte disabled
1 = 0E0000H — 0EFFFFH Low byte enabled

Bit 6 — Colour plane High access control

0 = 0E0000H -0EFFFFH High byte disabled
1 = 0E0000H -0EFFFFH High byte enabled

Bit 7 — Display/User RAM select

0 = 0D0000H — 0EFFFFH = user RAM
1 = 0D0000H — 0EFFFFH = Display Memory

Bit 8 — Not Used

Bit 9 — STOP LED

0 = on
1 = off

Bit 10 — POWER LED

0 = on
1 = off

Bit 11 — SHIFT LOCK LED

0 = on
1 = off

Bit 12 — DISK LED

0 = on
1 = off

Bit 13 — VOICE LED

0 = on
1 = off

Bit 14 — COLOUR SELECT LED

0 = on
1 = off

Bit 15 — CAPS LOCK LED

0 = on
1 = off

Port 24H — Centronics Status Port

This is a byte read only port.

Bit 0 — Busy

0 = Not Busy
1 = Busy

Bit 1 — Select

0 = Not selected
1 = selected

Bit 2 — Fault

0 = Fault
1 = No Fault

Bit 3 — Paper Empty

0 = Paper present
1 = Paper Empty

Bit 4 — Voice Controller LP23

Bit 5 — Disk Changed

0 = same disk
1 = disk changed

Bit 6 — Programmable Strap 1

Bit 7 — Programmable Strap 2

Interrupt Controller 2.3

Contents

Illustrations

Introduction

The Intel 8259A Programmable Interrupt Controller (PIC) is located on the Processor Board and forms the interface between the devices capable of generating interrupt requests and the interrupt control line of the 8086 processor. This is the same device as found on the Apricot pc/xi range of computers.

Interrupt requests are supplied to the PIC, when the associated device requires a service routine to be undertaken.

The PIC functions as the manager of the interrupt driven processing system; generating an active interrupt to the 8086 processor only when an incoming interrupt request has a higher priority than any interrupt service routine already in operation.

On acknowledgement from the CPU that it is able to accept the generated interrupt, the PIC supplies vectoring data to the CPU which acts as a pointer to the required service routine. The vectoring data, the mode of operation of the PIC and the priority of the interrupt requests are set up by the BIOS.

Details

General

The device connections to the interrupt request lines of the PIC are illustrated on the block diagram on Figure 1 and detailed in tabular format following. An active interrupt request is indicated by a logic high state on the appropriate request line.

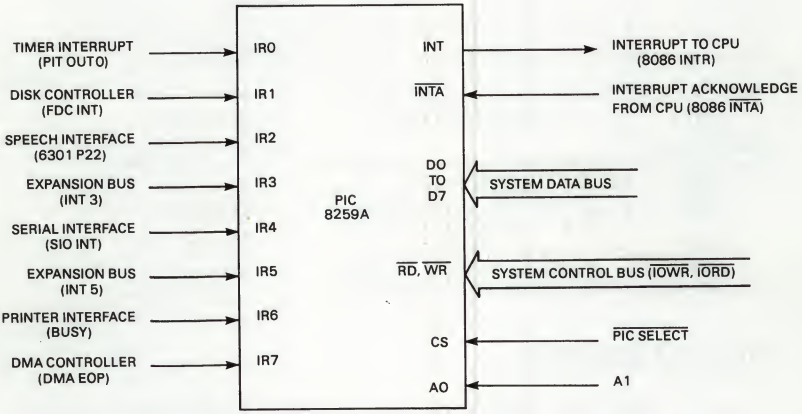


Figure 1. Interrupt Controller

PIC Pin Definition

IR0 to IR7	Interrupt Request inputs
INT	Interrupt output
INTA	Interrupt Acknowledge
D0 to D7	Data bus connection
RD	Read control line
WR	Write control line
CS	Chip select input
A0	System address bus input

Interrupt Sources

IR0	Timer (PIT OUT0)
IR1	Disk Controller (FDC INT)
IR2	Voice Controller (6301 P22)
IR3	Expansion 1 (EXP 1)
IR4	Serial Interface (SIO INT)
IR5	Expansion 2 (EXP 2)
IR6	Parallel Printer Interface (Busy)
IR7	DMA controller (DMA EOP)

The system software views the PIC as two input/output ports, with each port able to accept and provide a variety of data bytes. The port address locations, defined by the PIC select line and the system address bus connection, together with the abbreviations of the data bytes are given below. A detailed explanation of the data bytes can be found in the *Programming Considerations* section following.

Port Address	Data Bytes	Transfer Operation
68H	IRR/ISR	Read
68H	ICW1/OCW2/OCW3	Write
6AH	IMR	Read
6AH	OCW1/ICW2/ICW4	Write

Interrupt Sequence

The interrupt sequence, entered on receipt of a logic high on any of the interrupt request lines of the PIC is described in the following paragraphs. This sequence is the same regardless of the actual PIC interrupt request line being set active.

When one or more of the interrupt request lines are set into the active high state, corresponding bits in an 8-bit internal register within the PIC (the Interrupt Request Register - IRR) are also set.

The PIC selects the highest priority bit stored in the IRR, which is not masked by the software, for comparison with bits stored in a second 8-bit register (the Interrupt Service Register - ISR), to determine whether an interrupt should be issued to the CPU.

The Interrupt Service Register contains bits which indicate the interrupt service routines currently being executed by the CPU.

If the highest priority unmasked interrupt request is of a higher priority than the highest priority bit set within the ISR, the interrupt control line to the CPU is set into the active state.

If the highest priority unmasked interrupt request is not of a higher priority than the highest priority bit within the ISR, no further action takes place. The priority of the interrupt requests is determined by the priority mode selected.

Providing the software controlled interrupt-enable flag of the CPU is enabled, the CPU acknowledges the interrupt, by issuing two Interrupt Acknowledge pulses (INTA pulses) to the PIC.

The first INTA pulse latches the highest priority interrupt request from the IRR through to the ISR. The second INTA pulse enables vectoring data associated with the highest priority bit within the ISR onto the data bus.

The vectoring data (designated an interrupt number) is a single software defined byte, which is used by the CPU to specify the address location of the corresponding service routine.

On completion of the service routine, the associated ISR bit within the Interrupt Service Register is reset by the programmer sending a end of interrupt (EOI) byte to the PIC.

Interrupt Masking

Masking any of the interrupt requests is achieved under software control by utilising a third internal 8-bit register located within the PIC, known as the Interrupt Mask Register (IMR). The 8-bits within the IMR directly correspond to the 8-bits within the Interrupt Request Register. Setting an IMR bit prevents the PIC actioning any active state appearing on the associated interrupt request line.

Programming Considerations

General

Before the PIC can operate normally within the system, it has to be initialized with a series of command words which define the required operating mode, the vectoring data and the initial priority of the interrupt request inputs. This is set up by the BIOS at Boot time.

Once initialised, the operation of the PIC can be modified and controlled by a second series of command words which:

1. Enable the interrupt request lines to be individually masked.
2. Allow the ISR bits within the Interrupt Service Register to be cleared at the end of an interrupt service routine, to terminate the routine.
3. Enable the priority of the interrupt request lines to be changed.
4. Allow the status of the bits of the three internal registers within the PIC (IRR, ISR and IMR) to be analyzed.

The first series of command words are called Initialization Command Words and the second series, Operational Command Words.

Initialization Command Words

Three Initialization Command Words (ICWs) are required to set the PIC into an initial operating condition. The three words have to be issued in a fixed sequence, and if any changes to the initial operating condition are required, the whole sequence must be reprogrammed.

Once initialized; the priority of the interrupt requests are automatically assigned from IR0 (highest priority) through to IR7 (lowest priority); the PIC is able to accept interrupt requests.

The initialization programming sequence is ICW1 first, ICW2 next, and finally ICW4. ICW3 is not required since the system operates using a single PIC.

ICW1 defines two parameters when the PIC is used with an 8086 CPU. These are as follows:

1. The way the PIC senses an active interrupt request (either edge or level sensitive interrupt request).
2. Whether there is more than one PIC operating within the processing environment.

Due to the fact that the same interrupt request line from the Expansion Slot can be driven from more than one source (in effect in wire-ORed fashion), the level sensitive interrupt mode must be adopted. (In the edge sensitive mode, a transition on an interrupt request line from a device connected to the interrupt line would be undetected by the PIC, if the interrupt request line is already raised to logic high by the second device).

Since the PIC has to respond to a level sensitive interrupt request and there is only one PIC, the format of ICW1 is fixed as detailed below. The address location ICW1 is written to, is 68H in the system input/output space.

D7							D0	
x	x	x	1	1	x	1	1	*ICW1*

x indicates that the logic state is immaterial program to zero.

The function of ICW2 is to define a base address for the interrupt number. This is signified by five bits within the control byte. The three other bits required to form the whole interrupt number (the 3 least significant bits - LSB) are automatically inserted by the PIC and are dependent on the interrupt request line as detailed below.

IR	T2	T1	T0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

T* = Interrupt Number bit

The base address of the eight interrupt numbers assigned to the PIC is 50H as defined by the ICW2 format below. The I/O address location ICW2 is written to, is 6AH.

T7	T6	T5	T4	T3				
0	1	0	1	0	x	x	x	*ICW2*

The interrupt number associated with each interrupt request line is generated by combining the base address (the 5 most significant bits -MSB of the interrupt number) with the bits automatically inserted by the PIC (the 3LSB).

This assigns the following interrupt numbers to the interrupt request lines.

IR	Source	Type Vector
0	Timer	50H
1	Disk Controller	51H
2	Voice Controller	52H
3	Expansion 1	53H
4	Serial Interface	54H
5	Expansion 2	55H
6	Printer Busy Input	56H
7	DMA Controller	57H

The majority of the features provided by ICW4 are for use with systems utilising more than one PIC and therefore not relevant to the single PIC environment operated on the board. The command byte still has to be issued, to provide the PIC with the following information:

1. The associated CPU is an 8086 device.
2. Termination of an interrupt service routine is to be signified by software using an Operational Control Word, and not by the automatic end of interrupt facility (AEOI), available during the hardware interrupt acknowledge cycle. (AEOI is only suitable for systems with interrupts which occur at a predetermined rate).

The required format for ICW4 is as detailed below and is written to the address location 6AH in the system input/output space.

0	0	0	0	0	x	0	1	*ICW4*

Operational Command Words

After initialization the interrupt request lines are all operative with a decreasing order of priority from IR0 through to IR7 (i.e. IR0 highest priority, IR7 lowest).

The operation of the PIC can then be further controlled or modified using any one of three Operational Control Words (OCW1, OCW2 or OCW3). The OCWs are not dependent on each other and can be issued at any time during program execution.

Masking

OCW1 provides the facility for enabling/disabling individual interrupt request lines. This is achieved by issuing the control word to the 8-bit Interrupt Mask Register (IMR) within the PIC. The address location of the IMR is 6AH within the system input/output space and the format of OCW1 is as follows.

M7	M6	M5	M4	M3	M2	M1	M0	*OCW1*

Each bit within the IMR directly corresponds to an interrupt request line (M0 of the IMR affects IR0, M1 of the IMR affects IR1, M2 affects IR2 etc.).

Interrupt request lines are disabled (masked) when the corresponding bit within the IMR is set to logic high, and enabled (not masked), when the corresponding bit is set to logic low.

The status of the bits within the mask register can also be read by the programmer at address location 6AH within the I/O space.

Interrupt Termination

OCW2 is a dual purpose control word, which allows the priority of the interrupt requests assigned during the initialization sequence to be altered, and is also used to inform the PIC that an interrupt service routine is terminating.

The facilities provided by OCW2 for altering the priority of the interrupt requests are not required, since the actual hardware IR connections have been made on the basis of the priority assigned during the initialization routine.

The format of OCW2, used to inform the PIC that the interrupt service routine is at an end (enabling the PIC to reset the associated ISR bit within the Interrupt Service Register and other lower priority interrupts to be actioned), is as detailed below. The programmer must send this byte to the PIC following completion of the service routine to allow other interrupts to be serviced.

The address location OCW2 is written to, is 68H within the I/O space.

0	0	1	0	0	x	x	x

OCW2

Reading the ISR and IRR registers

OCW3 is a command word which provides more facilities to alter the priority of the interrupt request lines, and also allows the status of two internal registers within the PIC, the Interrupt Request Register (IRR) and the Interrupt Service Register (ISR), to be checked.

Facilities for altering the priority of the interrupt request lines are not required as explained above.

To read either of the two registers requires two operations to be carried out; a write operation using OCW3 to select the register; followed by a read operation to access the data within the register.

The format of OCW3 to select the registers is detailed below. The address location of OCW3 is 68H within the I/O space. The address location, the register data is read from is also 68H within the system input/output space.

x	0	0	0	1	0	1	RIS

OCW3

1 - Read ISR
0 - Read IRR

Display Control 2.4

Contents

Illustrations

Introduction

The display architecture of the Apricot Portable is a slightly different system to what is usually found on other microcomputers. It is totally graphics oriented (i.e. based on a bit-map instead of characters) and can also be software configured for a number of different resolutions and modes.

In both models in the current Portable range, the programmer has the facility to drive both the flat panel display (LCD module) and an optional colour monitor with different data and produce two different displays simultaneously.

Details

Portable Variations

The Portable does not use the more commonly found method of dual-porting the System RAM for display purposes (as found on the Apricot pc/xi range of micros). Instead it utilises a separate area of memory, designated as display RAM.

The base model of the Portable range is not as sophisticated as the other model since it is only fitted with enough display RAM (64 Kbytes) to operate in one of two modes. These are:

1. Driving the LCD module plus a colour monitor in any combination of two colours from sixteen (monochrome).
2. Driving a colour monitor using any combination of four colours from sixteen (LCD off).

The size of RAM is the only difference in display architecture between the base model and the second model within the Portable range, but as a result varies the features available to the programmer enormously.

The second model within the range incorporates the colour option circuitry, which is obtained by fitting an extra 64 Kbytes of DRAM to the Portables' internal circuitry. This is implemented as a factory fitted option.

The extra memory provides applications software with sufficient memory capacity to drive the flat panel display and an optional colour monitor in a variety of modes. The selected mode determines how much of the display memory is utilised. The modes are:

1. Driving the LCD module plus a colour monitor in any combination of eight colours from sixteen.
2. Driving the colour monitor using any combination of the full sixteen colours available (LCD off).

The 128 Kbyte area of display memory is mapped directly above 832K in the 1 Mbyte address range of the 8086 processor.

Display Modes and Features

The resolution of the LCD is fixed at 640 pixels wide by 200 pixels high. This is bit-mapped by a screen image located in an area of the display memory. The resolution is such that it allows the programmer to display the industry standard of 25 lines of text with 80 characters in each line, using characters formed in a 7 x 7 pixel matrix and contained in an 8 x 8 pixel cell.

The bit-map for the LCD is organised as a linear array of bytes in the display RAM, and thus easily allows the programmer to manipulate the display to mix text and graphics.

The colour modes, resolutions and display features available to the programmer for the base model are detailed in the next two paragraphs. The resolutions described match the resolutions of the current ACT colour monitors.

For USA usage and other countries using 60 Hz mains supply frequency:

1. 640 x 200 bit-mapped colour graphics using up to 2 colours + 640 x 200 bit-mapped LCD.
2. 640 x 200 bit-mapped colour graphics using up to 4 colours, LCD off.

For UK, European and other countries using 50 Hz mains supply frequency:

1. 640 x 256 bit-mapped colour graphics using up to 2 colours + 640 x 200 bit-mapped LCD.
2. 640 x 256 bit-mapped colour graphics using up to 4 colours, LCD off.

With a colour monitor connected, the programmer can select any permutation of colours from sixteen standard colours using a colour palette (any permutation of two colours from sixteen in the 2 colour modes, any permutation of four colours in the 4 colour modes).

The colour modes, resolutions and display features available to the programmer when the extra display RAM is fitted are detailed in the next two paragraphs.

For USA usage and other countries using 60 Hz mains supply frequency:

1. 640 x 200 bit-mapped colour graphics using up to 8 colours + 640 x 200 bit-mapped LCD.
2. 640 x 200 bit-mapped colour graphics using up to 16 colours, LCD off.

For UK, European and other countries using 50 Hz mains supply frequency:

1. 640 x 256 bit-mapped colour graphics using up to 8 colours + 640 x 200 bit-mapped LCD.
2. 640 x 256 bit-mapped colour graphics using up to 16 colours, LCD off.

With a colour monitor connected, the programmer can select any permutation of colours from sixteen standard colours using a colour palette (any permutation of eight colours from sixteen in the 8 colour modes, any permutation of sixteen colours in the 16 colour modes).

The palette is an area of I/O mapped RAM which determines the colour mix at the display outputs.

The basic difference between the two different scan line modes is one of quality of output as observed on the colour monitor. The higher resolution mode of 256 lines obviously produces a higher quality display due to the availibility of the extra scan lines.

In text-based applications, the 200 line mode is designed to be used with characters formed in a 7 x 7 pixel matrix and contained in an 8 x 8 pixel cell, (i.e. the same resolution font as the LCD module).

The 256 line mode enables a higher definition character font to be used with standard text-based applications. The font characters in this case are formed in a 7 x 9 pixel matrix and contained in an 8 x 10 pixel cell.

This enables both modes to be able to adopt the industry standard display format of 80 characters by 25 lines.

There is no hardware differentiation between text and graphics on either the LCD module or the colour monitor as previously mentioned; everything is graphics (pixel) based. i.e. A "dot" on the display screen (flat panel or colour monitor) is mapped by a corresponding bit (or bits) in the display memory.

In other words, it does not matter whether the Portable is displaying text or graphics, the display circuitry treats them both in an identical manner.

All the standard character attributes are available to the programmer for displaying text on the LCD. These are produced by direct bit manipulation of the character image in the display RAM rather than the more commonly used system of using attribute bits. Both normal and reverse video characters can be supported with any combination of the following attributes:

1. Underline.
2. Strikethrough.
3. Intensity (simulated by shadow printing).

The two models also have the facilities for driving the LCD module as described above, plus one added feature. A second image of the flat panel display can be stored in a spare location in the display RAM. This can be switched onto the LCD under software control instead of the first image.

As this feature allows fast display changes, it is particularly useful for applications, for example, that use Help menus. (The LCD can also overlap into an area of the RAM normally used for the colour display if the extra RAM is fitted, allowing up to four different images of the LCD to be switched in and out).

To obtain a sensible and usable "text mode" on the colour display for existing text based applications, attribute support by the BIOS is only provided in monochrome on the colour monitor (i.e. any two colours from the possible sixteen).

This allows the programmer to assign the same software character attributes to the colour monitor as to the LCD (normal and reverse video characters with any combination of; Underline, Strikethrough, Intensity). These are produced in exactly the same manner as for the LCD; via direct bit manipulation of the character image in the display RAM.

BIOS support for character attributes are not provided in the multi-colour modes due to the inherent nature of the colour display itself. Since the only effect an attribute is used for is to differentiate a character(s) from other characters, any of the standard attributes can easily be represented by assigning attributes to a different colour in a multi-colour mode, instead of the normal monochrome method.

Display Architecture

From a descriptive point of view, the display circuitry can be broken down into two different areas; the area which drives the LCD module and the area which drives the colour display. (This division is purely a convenient way of describing how the circuitry is programmed to drive the two different display technologies, and does not explain how the actual hardware circuitry is closely inter-related).

Driving the LCD Module

The LCD module is mapped by an LCD screen image in the display RAM.

LCD Bit-map

The resolution of the LCD module is fixed at 640 pixels wide by 200 pixels high. A single bit in the display RAM is required for mapping each pixel on the LCD. The screen image therefore occupies 16,000 bytes (640 x 200 bits).

A pixel is switched on if the corresponding bit in memory is set (logic 1), and off if the bit is not set (logic 0).

The bit map in RAM is a fixed image; i.e. a bit in a set location in the display RAM always corresponds to a fixed pixel-position on the LCD. There is no hardware circuitry provided for scrolling the LCD to alter the mapping — see below.

How the bit-to-pixel mapping is organised is illustrated in Figure 1.

The start address of the LCD image is at address location E8000H. The byte stored at this location corresponds to the eight pixels in the top left hand corner of the LCD in the first pixel line. The most significant bit (MSB) of the byte corresponds to the first pixel on the screen; the least significant bit (LSB), corresponds to the eighth.

All pixel-positions on the LCD are mapped by sequential bytes on even addressed boundaries only (i.e. E8000H, E8002H, E8004H, etc). The LCD image in memory thus equates to a linear sequence of bytes using low bytes only. (The high bytes are used for the colour display).

Eighty bytes are required to map each pixel line (and thus 80 x 200 bytes for the full display). The same sequence of high order bit to low order bit mapping is maintained throughout. (High order bits are mapped to the left, low order bits to the right).

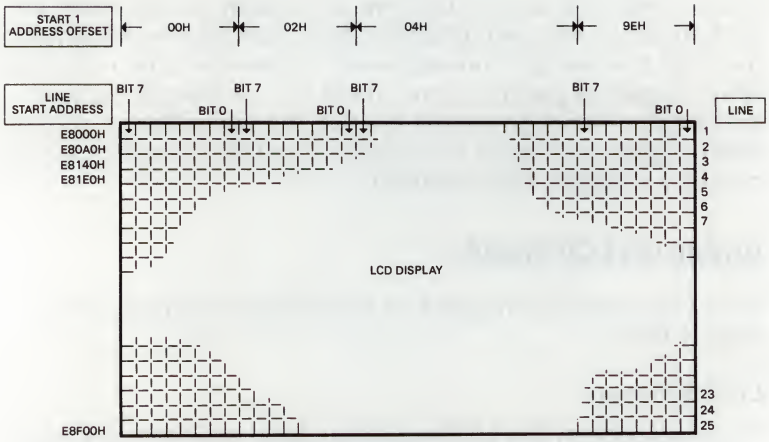


Figure 1. LCD bit map

Text Support Functions

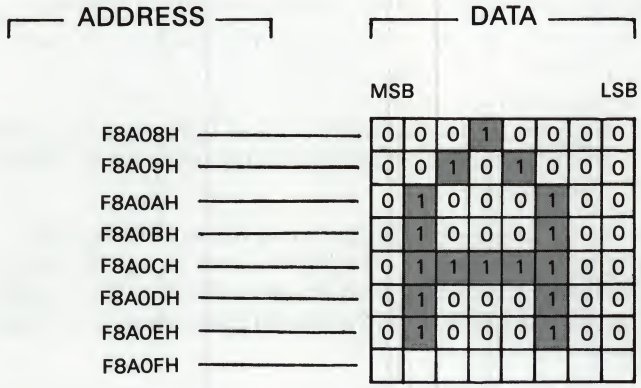
As the memory image is a bit-mapped version of the LCD, all normal text-oriented applications support features, i.e. a cursor and character attributes, have to be generated by direct manipulation of the images in memory.

Also, since there is no mechanism in the hardware for scrolling, scrolling has to be done purely in software. Because the memory image is a linear array of bytes aligned on even-addressed boundaries, scrolling can easily be done by moving the whole image using block memory to memory transfers (8086 instructions repeat move string — REP MOVS on words), and overwriting a section of the memory image to replace the part no longer to be displayed.

In order to generate the standard 80 character by 25 line text display, an 8 x 8 character font is used with the LCD. This is normally based upon characters constructed in a 7 x 7 matrix and contained within an 8 x 8 cell.

A default character font consisting of 128 characters (a standard ASCII set as illustrated by the first 128 characters in appendix C) is located in ROM. This ROM based set comprises eight bytes for each character and is mapped into contiguous memory locations, starting at absolute address F8800H.

The first addressed byte of a character in the font is the top character row, the second addressed byte is the second row, etc. The most significant bit of each byte represents the left edge of the character cell; the least significant bit the right edge, (see Figure 2).



Start Address of each character is F8800H + (8 x ASCII offset)

Figure 2. ROM based font character.

Another character font is normally down-loaded into system RAM at boot-time. This comprises 256 characters, all bounded within an 8 x 8 cell. BIOS support also allows other user-defined 8 x 8 fonts to be installed (see Screen Driver chapter for details).

Page Switching

An extra feature is available to the programmer for driving the LCD. This enhancement is provided purely by the availibility of the spare display RAM. It allows the programmer to produce fast display changes on the LCD by switching in and out different memory images (corresponding to an LCD page).

The programmer can build an LCD image in the background on a non-displayed page (e.g. a Help menu), and then simply switch in the new page when required. Up to four pages (one actively displayed on the LCD, three in the background) can be used in this way, but two of the pages overlap onto the colour screen memory in the 8 colour mode (see below).

These three extra memory images are mapped in different locations in RAM to the page described above in Figure 1, but are organised in a similar manner. One of the three extra images is mapped as a linear sequence of bytes starting from E0000H, and is also aligned on even addressed boundaries (i.e. low bytes only).

The other two images are mapped into the address range starting from D0000H and D8000H — see Figure 3 (and are only available on the model with the extra RAM). Both pages are defined by the low bytes (thus located on even addressed boundaries).

Page switching is controlled by simply toggling the state of a pair of bits within the word-wide control port mapped in the system I/O space at address location 2EH (bits 2 and 4). Changing to another page simply switches the start address of the LCD image to the value as specified by the control bits.

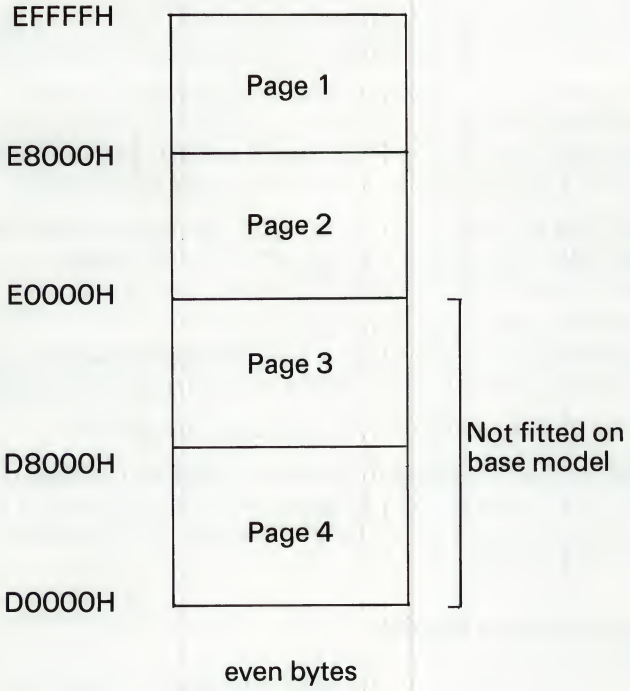


Figure 3. LCD Page Switching

The method of scrolling the LCD as described above is not as straightforward if the colour display is required for use at the same time. The repeat string operation (word REP MOVS) would normally adversely affect any image starting with the same address as the scrolled image but the one occupying the opposing byte within the word. (i.e. images which could be mapped onto a colour monitor).

To alleviate this potential problem and prevent it happening, a special high/low byte access control mechanism is available but is only available for the two LCD pages in the range E0000H to EFFFFH.

This mechanism alllows the programmer to switch the memory so that REP MOVS can be used by doing the standard word read/word write operation but the actual operation only performs word reads, byte writes. Only the LCD image is therefore affected by the repeat string operation.

This is controlled by another pair of programmable bits within the word-wide control port mapped in the system I/O space at address location 2EH (bits 5 and 6). See I/O port map in the System Details chapter for more information.

By varying the state of these two bits, the programmer has the choice on any word write operation in the display address range E0000H to EFFFFH of:

1. Enabling access to word locations.
2. Enabling access to the low byte address locations only.
3. Disabling access to the high byte address locations only.

Read operations, word or byte are totally unaffected.

The programmer also has the facility to switch the LCD on and off. This is achieved by toggling the state of another bit within the word-wide control port mapped at I/O address location 2EH (bit 3).

Driving the Colour Display

The programmer also has the facility to drive an optional colour monitor and the LCD simultaneously. The modes available for driving the LCD are as described in the section above.

The LCD and colour monitor are bit-mapped in two entirely different ways to match the required functionality of the two different display technologies. The LCD data is stored as a linear sequence of bytes on even addressed byte boundaries, as described above.

The colour monitor data is stored in a much more complex arrangement due to the inherent nature of the colour image as compared with the monochrome LCD image. This is based on a segmented structure. Before describing this, we will discuss other relevant aspects of the colour display architecture in order to gain an overview of the system adopted.

Colour Display Circuitry

Instead of being based simply on an area of RAM, where the programmer has only to take into account the bit-to-pixel mapping in memory, other programmable areas of circuitry are also built up around the colour display RAM (see Figure 4). This includes a programmable CRT controller (the commonly used Motorola 6845) and an area of I/O mapped RAM which is the colour palette.

The 6845 CRT controller is used for:

1. Accessing the data written into the display RAM for updating the image on the colour monitor.
2. Generation of timing signals which control the display scanning of the monitor.
3. Hardware display scrolling.

The colour palette selects the colour presented on the colour monitor as set by the logic state on the four colour coded display outputs (standard IRGB).

Each pixel on the screen of a colour monitor (the ACT colour monitors and other standard IRGB monitors) can be displayed in one of sixteen colours. In order to do this, the monitor has to be supplied with a 4-bit code sequence, to map each pixel. This is achieved by driving the four display outputs IRGB simultaneously.

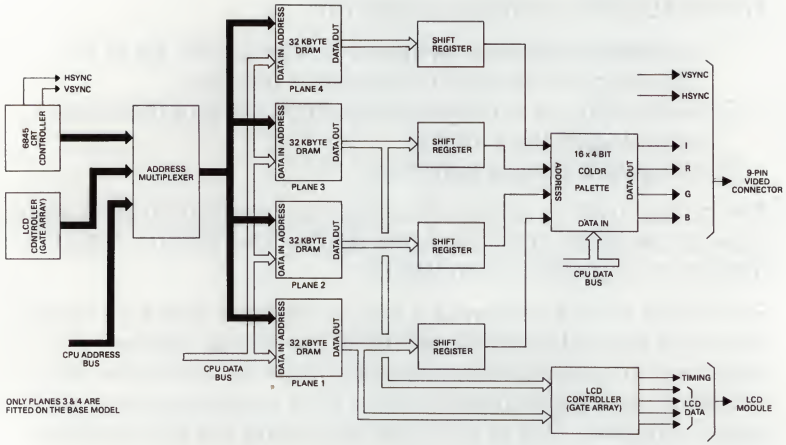


Figure 4. Display control block diagram

Memory Planes

The display memory organisation for driving the colour monitor is based upon four memory planes. (A memory plane is a block of memory in the display RAM, and in our case is not contiguous in memory — see Figure 5). Utilisation of the memory planes is entirely dependent on the mode selected.

Only two planes are available on the base model (planes 3 and 4). All four planes are available on the second model in the range.

In the 2 colour mode, the colour memory is normally mapped by plane 4 with plane 3 available for driving the LCD. This suits both models in the range.

In the full colour model in 8 colour mode, the colour memory is normally mapped by three of the planes (planes 1, 2 and 4) with the fourth plane (plane 3) available for driving the LCD.

In the 16 colour mode, the colour memory is mapped by all four planes and the LCD has to be switched off. (Since the LCD is switched on and off by an independent software switch, it is possible to route the data from the planes normally assigned for colour onto the LCD as well. But, since the colour memory image is of a different format, the LCD image will normally be incomprehensible).

The programmer can also use one of the colour planes (plane 1) for an extra two pages of LCD display information, if required, but this is at the expense of the number of colours able to be displayed. (The reduction to two planes for the colour monitor reduces the choice of colours to any four from 16).

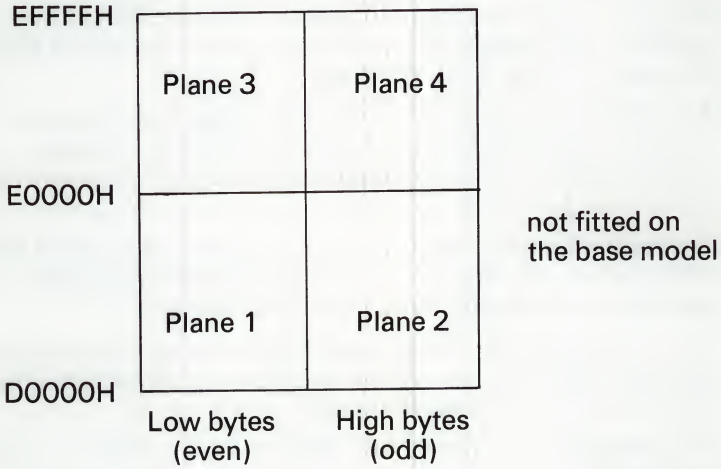


Figure 5. Memory Plane Organisation

Usage of the planes is entirely dependent on how many colours the programmer wishes to display on the monitor. 16 colours requires 4 planes, 8 colours requires 3 planes, 4 colours requires 2 planes, 2 colours (monochrome) requires a single plane. All other intermediate numbers of colours require the higher number of planes (e.g. 9 colours requires 4 planes).

The CRT controller is programmed to regularly access a byte of data from all four planes simultaneously (i.e. 32 bits), as the raster scan on the colour display is in progress (i.e. the active display period). The data accessed from the planes is supplied via four 8-bit shift registers to the colour palette.

The Palette

The programmer masks out the planes not in use (or the planes allocated to the LCD) and also selects the colours for display by programming the palette. The function of the palette is to translate the information from the planes into the colour coded IRGB outputs.

The palette is formed by a 16 x 4-bit static RAM, which is mapped into the system I/O address space by a single address (2AH). It can be regarded as a simple 16 entry table with 4-bits per entry.

The high nibble in a byte written to this address selects the entry within the table (i.e. forms the 4-bit RAM address), the low nibble programs the value into the table.

The palette is dual-ported being continuously accessed by the data from the four planes and also can be accessed by the programmer for setting up.

The translation of the data from the planes to colour-coded IRGB outputs is simply done by feeding the 4-bit parallel code sequence from the shift registers (1-bit per shift register at the pixel clock rate) to the four address inputs of the 16 entry palette RAM.

Data from plane 1 is supplied as the LSB address bit (bit 0) of the palette RAM, plane 2 data is supplied as address bit 1, plane 3 as address bit 2, and plane 4 as the MSB address bit (bit 3).

The 4-bit values programmed into the table set the colour selection at the output. If the programmer wants to mask out a plane (or planes), he does this by programming some of the entries in the palette with the same value. The entries which are programmed for masking are entirely dependent on the planes used for the colour display data.

This process of programming the palette is best illustrated by providing a few examples.

In order to display the full 16 colours on a colour monitor, all sixteen entries within the palette have to be programmed with different data, and all four planes filled with display data for driving the monitor.

The following table illustrates the required values for producing a 16 colour display.

Palette Address	Palette Value	Colour Output
0000	0000	black
0001	0001	dark gray
0010	0010	blue
0011	0011	light blue
0100	0100	green
0101	0101	light green
0110	0110	cyan
0111	0111	light cyan
1000	1000	red
1001	1001	light red
1010	1010	magenta
1011	1011	light magenta
1100	1100	brown
1101	1101	yellow
1110	1110	light gray
1111	1111	white

Each colour is set by writing the appropriate value to I/O address 2AH (e.g. To program the 15th entry to the colour selection for light gray, write EEH to 2AH).

The programmer does not have to program the palette in this order, but can map them in reverse sequence or in any other permutation as required.

The main advantage of the mapping shown in the table above is that the translation from display data written into the 4 planes to the codes on the IRGB outputs, is a straightforward one to one relationship. The 4-bit code accessed from the planes therefore directly correspond to the codes on the IRGB outputs, with each 4-bit code corresponding to a single pixel.

For an 8 colour display using planes 1, 2, and 4 only and masking out plane 3 (so that it can be assigned to the LCD), the programmer has to program the palette using eight different values only, repeating each value twice in certain positions within the palette table.

An example to illustrate the entries that have to be programmed with the same value to mask out plane 3 is detailed below. The actual colour values chosen are an arbitrary 8 from the 16 available.

Palette Address	Palette Value	Colour Output
0000	1111	white
0001	0010	blue
0010	1001	light red
0011	0110	cyan
0100	1111	white
0101	0010	blue
0110	1001	light red
0111	0110	cyan
1000	0001	dark grey
1001	0011	light blue
1010	0101	light green
1011	0000	black
1100	0001	dark grey
1101	0011	light blue
1110	0101	light green
1111	0000	black

If you examine any two values which are the programmed to the same colour value in the table above, you will see that the palette addresses of these two entries are the same apart from bit 2. This is the address line which is driven by the bit stream from plane 3.

The effect produced by placing the same colour value in this sequence is that any change of state made to data in plane 3 will be totally ignored. It is only the other planes which will produce a colour change as the bits from the planes change state. Changes to bits within plane 3 therefore will be in effect ignored, always producing the same colour output irrespective of the state of the data stored.

As an alternative, the LCD can be mapped onto plane 1, and the three other planes used for an eight colour display. The entries to be programmed with the same values in this case would be adjacent ones (i.e. 1 and 2, 3 and 4, 5 and 6, etc). Changes in state in plane 1 would then be ignored, allowing it to be programmed with display data for the LCD.

To produce a monochrome display on the colour monitor, the palette has to be programmed with eight values specifying one colour and eight values specifying the other colour. The order which the values appear in the table depends on the plane assigned to store the display data.

An example of programming the palette using plane 4 for a monochrome output is illustrated below.

Palette Address	Palette Value	Colour Output
0000	1111	white
0001	1111	white
0010	1111	white
0011	1111	white
0100	1111	white
0101	1111	white
0110	1111	white
0111	1111	white
1000	0100	green
1001	0100	green
1010	0100	green
1011	0100	green
1100	0100	green
1101	0100	green
1110	0100	green
1111	0100	green

This will produce a white pixel everytime the bit is not set in plane 4 and a green pixel when set (i.e. equating to a green on white display).

To utilise plane 1 as the monochrome display every second entry in the palette is programmed with the same colour value; e.g. 1, 3, 5, etc for one colour; palette entries 2, 4, 6, etc for the second colour. Both plane 2 and plane 3 can be used in the same manner, varying the programming of the palette accordingly.

Any combination of the 2 colours from 16 can be assigned to the colour monitor, by programming the palette with different values to produce a whole range of monochrome screens.

Because the display modes of four colours and less only use a maximum of two planes, it is possible to utilise the spare planes (depending on the usage for the LCD) for fast page switching effects on the colour monitor. This can be done by building up images in planes not in use, and then switching them in as required, by re-programming the palette.

Display on/off

The programmer can switch the display on the colour monitor by a programmable bit (bit 2) within the word-wide control port located at I/O address location 2EH. Setting the bit high enables data to be supplied to the colour monitor. Setting the bit low inhibits the display data.

CRT Controller/Scan Line Modes

The function of the CRT controller is to access the data stored in the memory planes, and generate the timing signals for controlling the raster scan of the colour monitor. It can also be used to "hardware" scroll the colour display.

The 6845 CRT controller is normally configured to match either one of two different scan line standards (the 200 line mode or the 256 line mode). This is achieved by programming the 6845 to match the line resolution required.

The different values programmed into the 6845 registers is one of the major differences between the two scan line modes. Another difference is the amount of memory utilised. The 256 line accesses more of the available display memory.

The controller employs exactly the same method in both the 200 and 256 line mode for:

1. Accessing the data from the planes.
2. Generating the raster scan timing signals (the frame rate is of course different for the two modes).
3. "Hardware" Scrolling.

Because of this, the following descriptive material equally applies to either configuration.

The CRT controller is not configured to access data from contiguous locations in RAM but on a four scan line repeat cycle. This produces a segmented memory structure for the memory map of the colour display.

It always accesses all four planes simultaneously, supplying 32 bits of data (two words — 1 byte from each plane) to the 4 shift registers which are connected to the palette RAM.

Each access cycle always:

1. Extracts a single word formed by two contiguous bytes, one byte from plane 1 and one byte from plane 2 (i.e. low and high bytes).
2. Extracts a single word formed by two contiguous bytes, one byte from plane 3 and one byte from plane 4.

The separation between the two words acccessed is the boundary value of 64K which separate planes 1 and 2 from planes 3 and 4. e.g. When the word stored at D0000H is accessed, the word stored at E0000H is also accessed. When the word stored at D0002H is accessed, the word at E0002H is also accessed.

The four scan line repeat cycle can be most easily explained by illustrating the process for a single plane. The same process occurs simultaneously on all other planes, the only difference is the addresses accessed.

The example described illustrates plane 1 and presumes that the CRT controller is set to access the first byte within each plane at the start of each frame scanning period (i.e. the display has been initialised to a start address of zero and has not been scrolled).

During the first active line period of each new frame period, the CRT controller circuitry generates a series of eighty sequential word addresses which access a linear sequence of 80 bytes starting at memory address D0000H. These 80 bytes from plane 1 (low bytes) are fed to the display output as a linear sequence of 640 bits to map the first scan line on the screen (see Figure 6).

In the next scan line sequence (line 2), the CRT controller generates another series of eighty sequential word addresses which access a linear sequence of 80 bytes starting at the memory address D4000H (i.e. the frame start address + 16K). These bytes are translated into a linear sequence of bits, which map the second scan line on the screen.

In the next scan line sequence (line 3), a linear sequence of 80 bytes is extracted out of plane 1 starting at the address location D8000H. (i.e. the frame start address + 32K).

In the next scan line sequence (line 4), a linear sequence of 80 bytes is extracted out of plane 1 starting at the memory address DC000H. (i.e. the frame start address + 48K).

In the next scan line sequence (line 5), a linear sequence of 80 bytes is extracted out of plane 1 starting at the memory address D00A0H (i.e. the frame start address + an offset of eighty word locations).

In the next scan line sequence (line 6), a linear sequence of 80 bytes is extracted out of plane 1 from the starting at the memory address D40A0H (i.e. the second line address + an offset of eighty word locations).

In the next scan line sequence (line 6), a linear sequence of 80 bytes is extracted out of plane 1 starting at the memory address D40A0H (i.e. the second line address + an offset of eighty word locations).

In the next scan line sequence (line 7), a linear sequence of 80 bytes is extracted out of plane 1 starting at the memory address D80A0H (i.e. the third line address + an offset of eighty word locations).

Line 2 data, line 6 data, line 10 data, etc, appears in contiguos locations on low byte boundaries only in the next 16K segment (see Figure 6).

Line 3 data, line 7 data, line 11 data, etc, appears in contigous locations on low byte boundaries only in the third 16K segment.

Line 4 data, line 8 data, line 12 data, etc, appears in contiguous locations on low byte boundaries only in the fourth 16K segment.

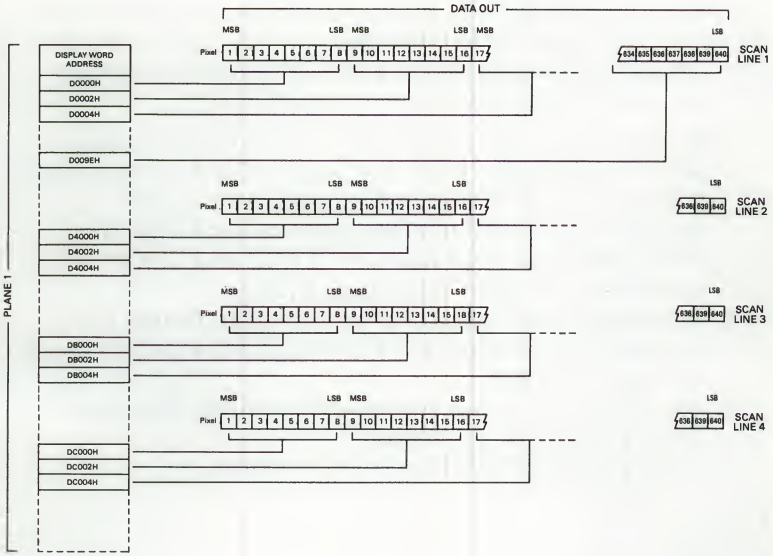


Figure 6. Colour Plane Segmentation

Colour monitor scrolling

Hardware scrolling is done by changing a start addresss in the CRT controller and rewriting the lines of data that require updating. All four planes are in effect scrolled simultaneously. The minimum increment for scrolling is four scan lines as defined by incrementing the start address by one.

CRTC Detail

General

The basis of the circuit for driving the colour monitor is the Motorola MC6845 CRT Controller (CRTC). Once initialised with operational parameters for the colour monitor (scan line rate, frame rate, mode, etc.), the CRTC:

1. Automatically and repetitively generates addresses which access data from the Display RAM to refresh the colour monitor screen (via a multiplexer, which re-orders and combines the address data from the CRTC with two clock signals, CLK90 and CLK45, to produce the 4-scan line access cycle).
2. Generates horizontal (line) and vertical (frame) sync pulses for controlling the scanning of the electron beam across the screen.
3. Generates an output signal which defines the active display period of the screen.
4. Allows the software to scroll the screen.

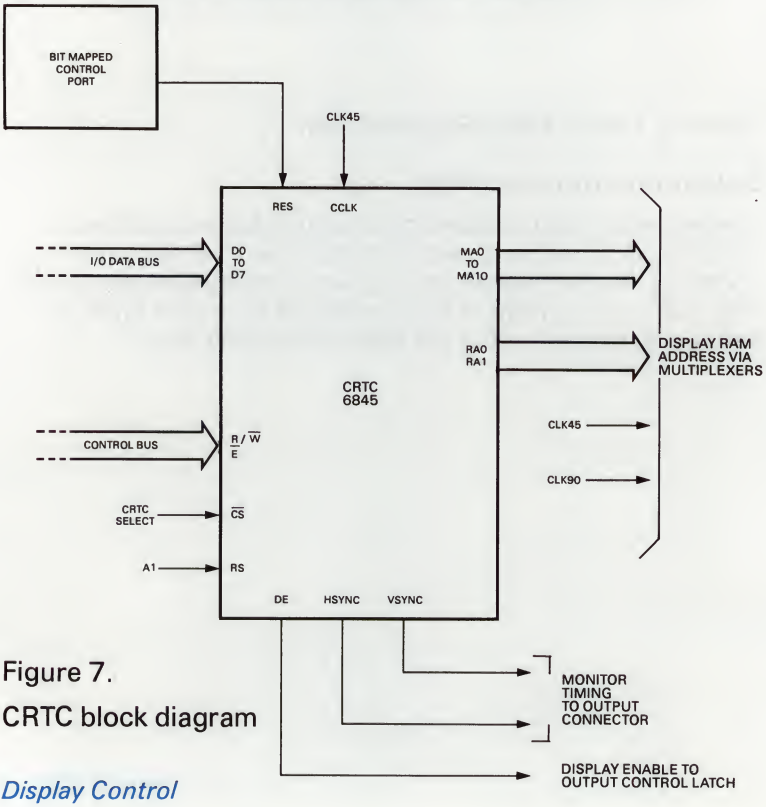


Figure 7.
CRTC block diagram

The refresh addresses generated by the CRTC are of two types, refresh address lines (MA0 to MA10), and raster address lines (RA0 and RA1). These are re-ordered and multiplexed to form the address lines to access the data stored in the Display RAM.

The CRTC is programmed to operate in a raster scan mode defined as non-interlaced. In this mode, the whole display raster on the screen is generated using a single frame scan.

The movement of the electron beam across the screen is controlled by the two outputs from the CRTC, Hsync and Vsync. Hsync controls the horizontal sweep of the beam across the screen during each frame (line scan rate). Vsync controls the vertical retrace of the beam back to the top of the screen at the end of each frame (frame rate).

The active display period on the screen is indicated by the Display Enable (DE) output, which is supplied to the display output stage. It enables the IRGB video signal outputs during the active periods, and inhibits them output during line and frame flyback periods.

Scrolling of the screen is achieved by writing a different start address into another pair of registers located internally within the CRTC. The start address specifies the first display memory address to be generated, at the start of each frame period. Internally the CRTC consists of a collection of registers, counters and comparators, which time all the logic activities (generation of addresses, sync. pulses, etc.) as the raster scan proceeds. The basic clock frequency used by the counters is CLK45.

CLK45 is derived from the 14 MHz dot clock and is at a frequency of 437.5 Khz.

Other registers within the CRTC are programmed with data for controlling the operation of the counters to determine various parameters for the Display Unit (e.g. active line period, vertical retrace period etc.). A definition of each of the registers is provided below.

Register Description

The CRTC contains 19 registers. 10 of the registers are programmed with operational parameters to match the Display Unit to the selected screen format (200 or 256 line mode). Two of the remaining registers define the start address of the scan. Another one of the registers specifies the address location of the other registers. Six of the registers are not applicable to the implementation of the CRTC within the circuit and are not used.

A summary of the registers is provided below. This is followed by a detailed description of each individual register. All registers can be written to and read from unless otherwise stated.

AR	Address Register	R9	Max scan line address
R0	Horizontal Total	R10	Cursor format (H) *
R1	Horizontal Displayed	R11	Cursor format (L) *
R2	HSync Position	R12	Start Address (H)
R3	Hsync Width	R13	Start Address (L)
R4	Vertical Total	R14	Cursor Position (H) *
R5	V. Total Adjust	R15	Cursor Position (L) *
R6	Vertical Displayed	R16	Light Pen (H) *
R7	VSync Position	R17	Light Pen (L) *
R8	Interlace Mode		

* indicates registers not used

Address Register

5-bit register located at 6CH in the system I/O space and can be written to only. Acts as the pointer for the other control registers.

Prior to any data transfer between the processors and any of the 18 control registers, the source/destination register has to be defined by programming the address register with the address pointer. The pointer for each control register is the binary equivalent of the decimal "R" number. e.g. Interlace Mode control register R8 is specified by programming the address register with 08H.

Data is then written into the specified register by an I/O write to address location 6EH.

Horizontal Total Register (R0)

The contents of this register determine the line frequency of the HSync output and is specified in CLK45 periods (437 KHz). The actual number is equivalent to the total number of clock periods in the active scan line time and the line retrace period minus one (active + retrace — 1 clock period).

Horizontal Displayed Register (R1)

This register specifies the number of clock periods in each active scan period.

HSync Position Register (R2)

The contents of this register specify the position of the horizontal sync pulse relative to the start of the active scan line period. The value is programmed in CLK45 periods. The effect of increasing the value in R2 is to shift all characters displayed on the screen to the left. Decreasing the value shifts the whole character display to the right.

HSync Width Register (R3)

The contents of R3 set the width of the Hsync pulse in units of the CLK45 clock period.

Vertical Total Register (R4)

This register with R5 define the frequency of the VSync pulses. R4 is programmed with a value, in character row periods (i.e. 16 scan line periods), just less than the desired frequency. The fine adjustment to achieve the exact VSync frequency is provided by R5. The actual value of R4 is one less than the desired number of character row periods.

Vertical Total Adjust Register (R5)

This register provides the fine control of the frequency of VSync and is programmed in scan line periods.

Vertical Displayed Register (R6)

The contents of this register determine the number of displayed character rows on the screen.

VSync Position Register (R7)

The contents of this register specify the position of the vertical sync pulse relative to the start of the first active scan line period. The value is programmed in Character row periods. Increasing the value in R7 shifts the whole display on the screen upwards. Decreasing the value shifts the display downwards.

Interlace Mode Register (R8)

This register selects the raster scan mode. A choice of non-interlace (00H), interlace (01H), or interlace sync and video (03H) modes are available. In the non-interlace mode, the scan lines are refreshed at the frame rate. In the two interlace modes, the frame is divided into two fields, an odd field and an even field. In the interlace mode, the same information is displayed in both fields. In interlace sync and video, the even lines of a character are displayed in the even field, and the odd lines in the odd field.

Max Scan Line Address Register (R9)

This register determines the number of scan lines per character row and is programmed with a number one less than the desired value.

Cursor Format (Start) Register (R10)

Not used.

Cursor Format (End) Register (R11)

Not used.

Start Address Register (R12)

The contents of this register with the contents of R13 define the first refresh address of each new frame period. R12 is programmed with the 6 most significant bits of a 14-bit word. The 8 least significant bits are provided by R13.

Start Address Register (R13)

See R12 above.

Cursor Position Register (R14)

Not used.

Cursor Position Register (R15)

Not used.

Light Pen Register (R16)

Not used.

Light Pen Register (R17)

Not used.

Initialising the CRTC

The values supplied to the control registers to initialise the CRTC to operate with the Portable Colour monitors for the two different display modes are detailed below. The control registers are programmed using two separate write operations.

The first write is to address location 6CH in the system input/output space (the address register) to specify the control register pointer address. The second write is to address location 6EH in the system input/output space, which loads the value into the control register, specified by the pointer.

256 line Modes

Register	Value	Register	Value
R0	3B	R8	03
R1	32	R9	0E
R2	30	R10	00
R3	0C	R11	0F
R4	19	R12	00
R5	0A	R13	00
R6	19	R14	00
R7	19	R15	00

200 line Modes

Register	Value	Register	Value
R0	3B	R8	03
R1	32	R9	0E
R2	30	R10	00
R3	0C	R11	0F
R4	19	R12	00
R5	0A	R13	00
R6	19	R14	00
R7	19	R15	00

CRTC Connections

D0 to D7	Data bus. Used to transfer data between the internal registers of the CRTC and the processors.
R/$\overline{W}$	Read/write control input connected to the DT/$\overline{R}$ control line of the system control bus. Used in conjunction with $\overline{CS}$, RS and E to control data transfers between the CRTC and the processors. R/$\overline{W}$ determines the direction of data transfer; logic high from the CRTC (read), logic low to the CRTC (write).
$\overline{CS}$	Chip Select. Address input. Active state, logic low. When active indicates that the CRTC is selected for a data transfer operation.
RS	Register Select. Input connected to A1 of the system address bus. Used to select either the address register (logic low) or one of the eighteen control registers (logic high).
E	Enable input. Data strobe signal for latching data to/from the data bus. Logic high to low transition at the end of the second processor clock cycle after the address is valid.

$\overline{\text{RES}}$	Input signal to reset the CRTC. Active state, logic low. Generated by writing to a word-wide control port (I/O address 2EH — bit 0). When active, all counters within the CRTC are cleared and all outputs are forced to logic low. The contents of the registers are unaffected.
CCLK	Character clock. Input signal (CK45) generated from an on-board timing circuit, with a 50% duty cycle, derived from the 14 MHz dot clock. Used as the basic CRTC timing signal with a frequency of approximately 0.45 Mhz.
MA0 to MA10	Refresh memory address signals. Address outputs, which change every character clock period during the active scan line period to access the Display RAM and thus refresh the CRT screen.
RA0 to RA1	Raster address signals. Address lines supplied to the Display RAM and combined with the Refresh memory addresses to access the display data.
VSYNC	Sync pulses supplied to the Display Monitor to control the retrace of the electron beam back to the top of the screen at the end of each active frame period. Pulses are generated at a rate of approximately 50 Hz — 256 line mode, 60 Hz — 200 line mode.
HSYNC	Sync pulses supplied to the Display Unit to control the horizontal sweep of the beam across the screen during the active frame period. The pulses are generated at a scan rate of 15.625 kHz.
Display Enable	Output generated by the CRTC which defines the active san line/retrace periods of the electron beam. Logic high indicates the active periods, logic low the retrace periods.

Expansion Slot 2.5

Contents

Illustrations

Introduction

The Portable Expansion Slot is located on the Processor Board. It provides an extension of the processing system for use by optional Expansion boards. Access to the slot is made by removing the Expansion cover on the back of the Systems Unit.

Details

Design philosophy

A high degree of compatibility has been maintained with the other products within the Apricot pc/xi range of computers. This is such that all existing ACT Expansion boards (Winchester Controller, Modem, RAM cards, etc) can be used within the Portable without any modification to the Expansion Board hardware.

The philosophy for using multiple Expansion boards is different to the one originally adopted on the Apricot pc/xi range of machines. Two approaches are available to the user for increasing the capabilities of the Portable via the Expansion Slot if he has more than one Expansion board.

The first approach is to swap boards as and when required. The plastics for the Portable have been specially designed to allow easy access to the machine for interchangeability of Expansion boards. (The expansion cover is removed by applying slight downwards pressure to the top of the cover panel to unhook the top retaining clips and then pulling the cover away in an upwards direction).

The potential problem of applications software requiring more RAM when using a certain Expansion board is alleviated by the availability of the on-board RAM expansion feature of the Portable (i.e. Two sizes of main memory are available as a manufactured option - 256K or 512K).

The second approach is to extend the expansion bus to a separately powered Expansion Unit fitted with multiple Expansion Slots. The Expansion Unit is responsible for re-powering the Expansion bus to meet the drive capability of multiple Expansion Slots.

Expansion Slot Detail

The extension connections wired to the Expansion Slot are:

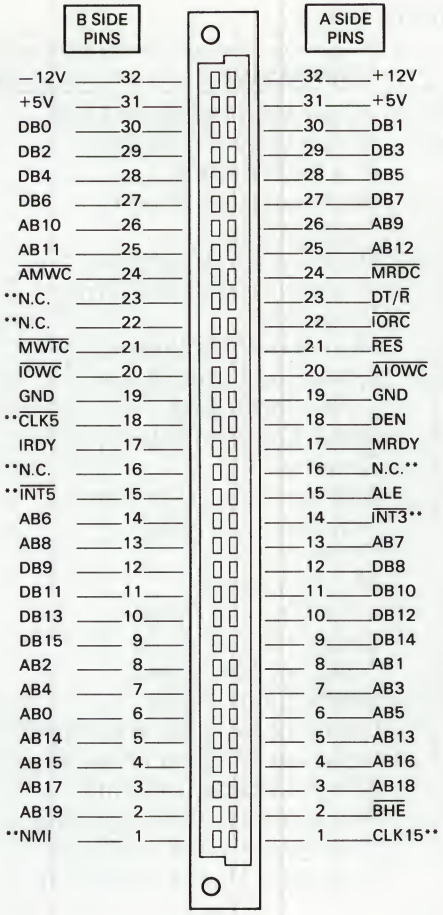
1. The 16-bit system data bus.
2. The 20-bit system address bus.
3. Various control and timing signals.
4. Power supply outputs.

The same physical connector is used for the Expansion Slot as used on all current machines within the Apricot pc/xi product range. This is a 64-way connector (DIN 41612, 2 by 32 female, with a type B housing).

Of the 64 connections routed to the slot, 60 are as on the pc/xi range of micros and are compatible connections. There are minor differences in detail. For example, Interrupt 2 on the pc/xi range is Interrupt 3 on the Portable - see Figure 1 for further details.

The four connections that are substantially different are also marked on Figure 1. The main area of difference is that DMA facilities are not routed to the Portable Expansion Connector as provided on the Apricot pc/xi range, these pins are left open circuit.

The reason for dropping the DMA facilities is the incompatibility with the DMA controllers used on the Apricot machines. The Portable uses an Intel 8237; the pc/xi range, the Intel 8089 IOP; the Apricot F1 has no DMA at all.



**Indicates differences with other products within the Apricot range. See Table below.

Pin	pc/xi	Portable	F1
A1	CK15	CK15	CK15 (14MHz)
A11	INT2	INT3	INT2
A16	EXT2	N.C.	N.C.
B1	NMI	NMI	N.C.
B15	INT3	INT5	INT3
B16	EXT1	N.C.	N.C.
B18	CK5	CK5	CK5 (4.67MHz)
B22	DMA1	N.C.	N.C.
B23	DMA2	N.C.	N.C.

Figure 1. Expansion Connector

Pin Definition

Pin	Description	Input/Output
AB0 to AB19	20-bit system address bus	Output
DB0 to DB15	16-bit system data bus	Bi-directional
$\overline{BHE}$	Bus High Enable	Output
ALE	Address Latch Enable	Output
$\overline{DEN}$	Data Enable	Output
DT/$\overline{R}$	Data Transmit/Receive	Output
$\overline{AMWC}$	Advanced Memory Write Command	Output
$\overline{MWTC}$	Memory Write Command	Output
$\overline{AIOWC}$	Advanced Input/Output Write Command	Output
$\overline{IOWC}$	Input/Output Write Command	Output
$\overline{MRDC}$	Memory Read Command	Output
$\overline{IORC}$	Input/Output Read Command	Output
MRDY	Memory Ready	Input
$\overline{IORDY}$	Input/Output Ready	Input
$\overline{RES}$	System Reset	Output
CLK15	15MHz Clock signal	Output
$\overline{CLK5}$	5MHz Clock signal	Output
$\overline{INT3}$	Interrupt Request (Priority 3)	Input
$\overline{INT5}$	Interrupt Request (Priority 5)	Input
NMI	Non-Maskable Interrupt	Input
+12V	Processor Board supply rail	Output
−12V	Processor Board supply rail	Output
+5V	Processor Board supply rail	Output

Electrical specification

Current Consumption

Maximum allowed current consumption of a circuit board fitted into the Expansion slot is:

0.5A from the +5V rail.
50mA from the +12V and —12V rails.

Signal Outputs

All signal outputs (data, address, control and clocks) have the capability to drive a maximum of 2 LS TTL loads, i.e.

Logic high state voltage (Voh);

2.0 < Voh < 5.25 with maximum high state output source current of 40 μA.

Logic low state voltage (Vol);

— 0.5 < Vol < 0.8V with maximum low state output sink current of 0.8mA.

Signal Inputs

All signal and control inputs (apart from the interrupt lines) require a tri-state driver stage meeting the following requirements.

Logic high state voltage (Voh);

2.4 < Voh < 5.25V with maximum high state output source current of 400μA.

Logic low state voltage (Vol);

— 0.5 < Vol < 0.5V with maximum low output state sink current of 8mA.

The interrupt inputs require to be driven by an open collector driver stage. These lines on the CPU and Display Board are fitted with pull-up resistors (3.3k).

Pin Detail

A description of each connection to the slot is detailed below.

DB0 to DB15	16-bit system data bus. Connected to the pair of transceivers which form the interface between the processor and the system data bus. DB0 is the LSB, DB15 the MSB. In effect, the bus is divided into two parts (the low order section DB0 to DB7, and the high order section DB8 to DB15), to support both 8-bit (byte) and 16-bit (word) data transfers. Byte transfers from/to even address locations are transferred on the bus lines DB0 to DB7 and byte transfers from/to odd address locations are transferred on bus lines DB8 to DB15. Word transfers from/to even addresses are transferred on bus lines DB0 to DB15 in a single operation. Word transfers from/to odd address locations are automatically transferred in two consecutive data byte transfer operations; the first operation uses bus lines DB8 to DB15 (odd address transfer), and the second operation use bus lines DB0 to DB7 (even address transfer).
AB0 to AB19	20-bit system address bus. Connected to the octal D-type latches which demultiplex the 20 address bits from the local bus of the processor (time multiplexed address/data bus for the 16 LSB and the time multiplexed address/status bus for the 4 MSB). AB0 is the LSB, AB19 the MSB. AB0 has a special function and is normally used in conjunction with the BHE signal to condition circuitry for byte or word data transfers.

$\overline{BHE}$	Bus High Enable. Connected to a D-type latch which demultiplexes the BHE signal from the local bus of the processor. Normally used in conjunction with AB0 to enable circuitry for byte or word data transfers on the low or high order sections of the 16-bit data bus as follows;

$\overline{BHE}$	AB0	Transfer operation
0	0	Whole Word
0	1	High order byte
1	0	Low order byte
1	1	None

ALE	Address Latch Enable. Connected to PAL 1 on the CPU Board and driven by an S04 inverter. Negative edge of the active high pulse provides an indication of when the address is valid.
DEN	Data Enable. Output from PAL 1 on the CPU Board. Active high during memory and input/output data transfers.
DT/$\overline{R}$	Data Transmit/Receive. Output determined by the state of the read pulse generated either by the processor or the DMA controller. Signifies direction of data flow. Logic high indicates data transmission from the processing system; logic low, data reception.
$\overline{AMWC}$	Advanced Memory Write Command. Active low control signal which is set active before the Memory Write Command to provide memory-mapped devices an earlier indication of a write cycle.
$\overline{MWTC}$	Memory Write Command. Active low write command for memory-mapped devices.
$\overline{AIOWC}$	Advanced Input/Output Write Command. Active low control signal which is set active before the Input/Output Write Command to provide input/output devices an earlier indication of a write cycle.
$\overline{IOWC}$	Input/Output Write Command. Active low write command for input/output devices.

$\overline{\text{MRDC}}$	Memory Read Command. Active low read command for memory-mapped devices.
$\overline{\text{IORC}}$	Input/Output Read Command. Active low read command for input/output devices.
MRDY	Memory Ready. Input connected to the 8284A Clock Generator (RDY1) via an LS21 gate. Normally at logic high, but is set low to command the processor to extend the control transfer commands, by inserting wait states, until the selected memory-mapped device is ready for the data transfer operation. MRDY returning to logic high indicates that the selected memory-mapped device is ready for the data transfer operation (read or write).
IORDY	Input/Output Ready. Input connected to the 8284A Clock Generator (RDY2) via an LS21 gate. Normally at logic high, but is set low to command the processor to extend the control transfer commands, by inserting wait states, until the selected input/output device is ready for the data transfer operation. IORDY returning to logic high indicates that the selected input/output device is ready for the data transfer operation (read or write).
$\overline{\text{RES}}$	System Reset. Output from the 8284A Clock Generator via an inverter. Active low state generated by the Clock Generator on receiving a hardware reset (power on reset or via the Reset button on the Keyboard Unit).
CLK15	15 MHz Clock signal. Buffered output from the 8284A Clock Generator with a 50% duty cycle.
$\overline{\text{CLK5}}$	5 MHz Clock signal. Output from the 8284A Clock Generator via an inverter. Inverted form of the clock signal supplied to the processing elements. 66% duty cycle.

$\overline{\text{INT3}}$	Interrupt Request (Priority 3). Input line connected to the Interrupt Request 3 input of the Programmable Interrupt Controller (PIC) via an inverter. The interrupt type number supplied to the 8086 processor on acknowledgement of the interrupt request is 53H. (The interrupt type number acts as a pointer to the interrupt service routine).
$\overline{\text{INT5}}$	Interrupt Request (Priority 5). Input line connected to the Interrupt Request 5 input of the PIC via an inverter. The interrupt type number supplied to the 8086 processor on acknowledgement of the interrupt request is 55H. (The interrupt type number act as a pointer to the interrupt service routine).
$\overline{\text{NMI}}$	Non-Maskable Interrupt. Input connected to the NMI input of the 8086 processor via an inverter. A logic high to low transition on NMI generates the predefined interrupt type number 2 internally within the 8086, which acts as a pointer to an interrupt service routine.

Address Allocation

The available address locations in the system memory space and input/output space allocated to the Expansion Slots are detailed in the following paragraphs. 8-bit devices connected to the lower half of the data bus must be located on even address boundaries and 8-bit devices connected to the upper half, on odd address boundaries.

System memory

The available address locations in the system memory space is dependent on the model within the Apricot Portable range.

40000H to D0000H - 256 Kbyte Portable *
80000H to D0000H - 512 Kbyte Portable *

* Expandable to F0000H using the software switch for switching between display and system RAM.

1 processor wait state is automatically inserted on all memory read and writes. The processor wait states can be extended if the MRDY input to the system is utilised.

System input/output space

The available I/O address range for the Expansion Boards is split into a number of categories according to the type of board. This approach has been adopted in-house within ACT for its whole range of Expansion products to avoid the possibility of the Expansion Slots being filled with Expansion Boards utilising the same I/O addresses.

Any third party vendor who wishes to design Expansion Cards which are/will be compatible with ACT's existing and future products should adhere to the same scheme.

The following table details the I/O addresses assigned to the various categories of Expansion Cards. (The values of the port addresses are in hexadecimal).

Expansion Board Category	I/O Port Address	I/O Port Address
Colour/Graphics Boards	80 to 8F	100 to 10F
IEEE 488 Controllers	90 to 9F	110 to 11F
Local Area Networks	A0 to AF	120 to 12F
Gateways	B0 to BF	130 to 13F
Miscellaneous Communications	C0 to CF	140 to 14F
ACT Reserved	D0 to DF	150 to 15F
Winchester Boards	E0 to EF	1E0 to 1FF
Modems	F0 to F7	1C0 to 1DF
Undefined	—	160 to 17F
Undefined	—	180 to 19F
Undefined	—	1A0 to 1BF

1 processor wait state is automatically inserted to any I/O read or write by accessing any of these addresses. The processor wait states can be extended if the IORDY input to the system is utilised.

Using Interrupts

To account for the possibility of two boards both requiring the use of the same interrupt request line:

1. All boards using either of the interrupt lines must contain a status register.
2. The interrupt must be driven by an open collector output.
3. The software device driver must operate in the manner described in the paragraphs below. The use of the status register will also become apparent.

If space permits on the board, the designer should track the board to both interrupt lines and provide a link to provide the board with the option of driving either interrupt line.

External Expansion Interrupt Set-up

To allow the software programer to easily link his device driver into all machines in the Apricot range of computers (Portables, F1 and pcs/xis with the generic BIOS), a special software interrupt is available. The purpose of this is to allow the programmer to mask out the differences in interrupt structure between the various machines within the Apricot product range with regard to interrupt type vectors.

The interrupt routine produced by generating an active interrupt on INT3 on the Portable is identified by the interrupt type number 53H. The interrupt routine produced by generating an active interrupt on the corresponding interrupt line on the Apricot F1 is identified by interrupt type number 60H.

The Expansion set up interrupt 0F4H enables the programmer to install his vectors in the appropriate locations by relating them to the interrupt line without regard to the particular Apricot machine. (This applies to Apricot Portables, Apricot F1s and Apricot pcs and xis fitted with either the ROM BIOS or the RAM BIOS equivalent).

To use the interrupt requires the programmer to load the 8086 registers AL, BX and CX with the data as shown below and then generate the interrupt. Data is returned in BX and CX as shown. The purpose of this is explained in the next few paragraphs.

0F4H - External Expansion Interrupt Set-up

Input: AL = 0 - Set External Interrupt 3 (Winchester) *
AL = 1 - Set External Interrupt 5 (general) *
BX = Vector Offset Word
CX = Vector Segment Word

Output: BX = Old Vector Offset Word
CX = Old Vector Offset Segment

* See Figure 1 for corresponding Interrupt Lines on the Apricot F1 and the Apricot pc/xi range of machines. The interrupt line INT3 on the Portable corresponds to INT2 on the Apricot pc/xi range of machines. This was originally reserved for use solely by the Apricot Winchester Controller. This is no longer a requirement if machines are fitted with Revision 9 or later versions of Winchester Controller Board.

Installing the device driver at system boot

One of the routines executed during initialisation must be to install the two vector words associated with the device driver service routine using the Expansion Set-up interrupt. The old vector addresses returned must be saved (the reason for doing this is given below).

Run-time operation

The first task undertaken by the service routine must be to read the status register to check that the device is the actual cause of the interrupt. If not, the device driver should immediately relinquish control to the service routine specified by the vectors returned by the Expansion Set-up Interrupt.

If the interrupt is caused by its own device as indicated by the status register, the driver naturally performs the appropriate service routine. At the end of the routine, the device driver must relinquish control to the routine specified by the vectors returned by the Expansion Set-up Interrupt.

This effect, where multiple device drivers are installed all using the same interrupt line, is similar to the technique of "daisy chaining" interrupt lines and acts as a logical extension to the MS-DOS 2.0 installable device driver philosophy.

Notes

1. At boot-time the BIOS assigns in effect a null device driver to the available interrupt numbers, which performs two functions:

   a. Supplies the end of interrupt command to the Interrupt Controller.
   b. Returns control back to the program point of interruption.

   The "null device driver" is always the last driver to be serviced, since it will always be the first one installed (i.e. the end of the "daisy chain").

2. Since the last loaded driver will always be the first device serviced following an interrupt (i.e. the beginning of the "daisy chain"), the order of loading multiple device drivers automatically assigns an order of priority. Time critical device drivers therefore, should always be loaded last.

3. The maximum time allowed for each interrupt service routine should generally not exceed 10ms.

Expansion Board Layout Detail

The dimensions and layout details for an Expansion Board are detailed on Figure 2.

The uppermost view illustrates the overall board dimensions and the location of the connector.

The middle projection provides a different view of the connector and details the maximum height available for components mounted on the board.

The lower illustration details all the drilling requirements for the printed circuit board and the board area available.

This design of Expansion Board is for a standard board which will fit into the Expansion Slots on the dual expansion slot version of the **Apricot pc/xi** range of micros. These are currently the machines which are most restricted in terms of space available for Expansion components.

In order to design boards which encompass the full range of Apricot products, all boards should be designed to these tighter tolerances. (In general, all Expansion Boards should be designed to the tighter tolerances specified for the slot Expansion 1 on the **Apricot pc/xi** range to allow other boards which may require the extra depth of Expansion 2 slot to fit into the pc/xi machines).

Note:

1. The 3.85 mm diameter holes located in three corners of the board are only required on expansion cards made up of two boards, in the form of a "sandwich" (i.e. a main board fitted with the expansion connector, and a piggyback board separated from the main board by spacers). In this situation, the three holes can be used as general purpose tooling holes and also as screw holes for the spacer fixings.
2. On single board expansion cards, the 3.85 mm diameter located in the upper right-hand corner is not required, thus providing a small extra area of board space. The two 3.85 mm holes on the left-hand side of the board together with the connector fixing hole in the lower right-hand corner can then be used as tooling holes, if required.

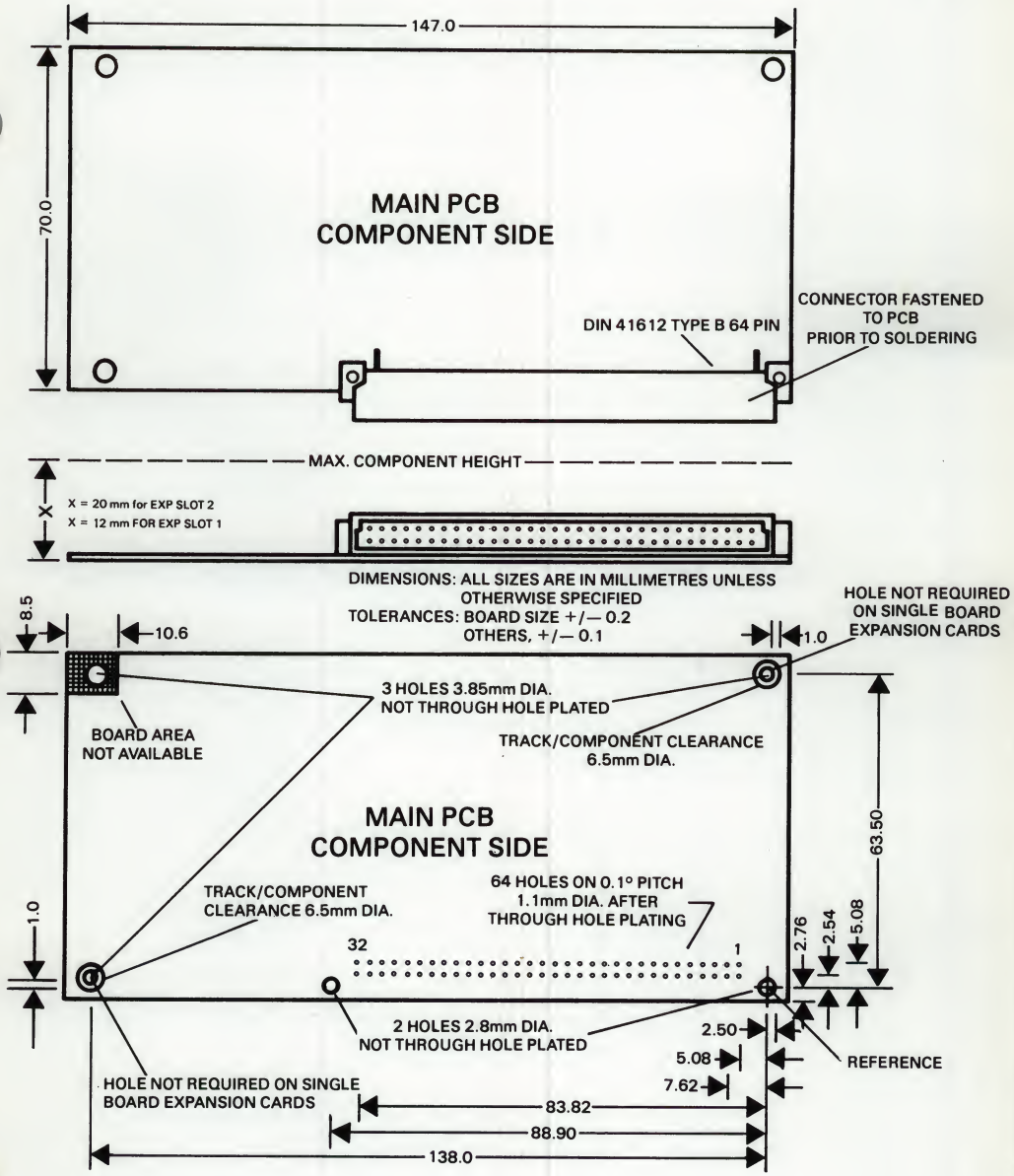


Figure 2. Expansion board detail for ***pc/xi*** range

Floppy Disk Interface 2.6

Contents

Illustrations

Introduction

The Floppy Disk Interface is located on the Interface Board and consists of the elements of circuitry as illustrated on the block diagram Figure 1. The interface provides all the control functions necessary for formatting and transferring data to/from MicroFloppy Disks via a Microfloppy disk drive.

The disks are soft-sectored and encoded with the same disk format as the double-sided disks used on the Apricot pc/xi range of products. This format is logically developed from the IBM system 34 format (a standard format for 8 inch disks), and is specifically designed to obtain the optimum number of data bytes on a double-sided 3.5 inch MicroFloppy disk.

The format employs double density MFM coding with 512 bytes per sector and 9 sectors per track. The number of tracks per side of disk is 80.

A brief description of the disk format is included at the end of this section (under the heading Track Format). This should be read now, if the reader is unfamiliar with the method of recording data on disks.

Control connections between the interface and the drive is made internally within the Systems Unit. The Floppy Disk Drive Connector is a 26-way male IDC terminal mounted on the component side of the Interface Board. An associated ribbon cable assembly links the interface to the drive.

Regulated power supply voltages for the disk drives are supplied directly from the Power Supply Unit via a 4-wire cable assembly.

Details

General

The Floppy Disk Interface consists of a Western Digital WD2797-02 Floppy Disk Controller (FDC), a series of buffers and control ports, and the interface connector.

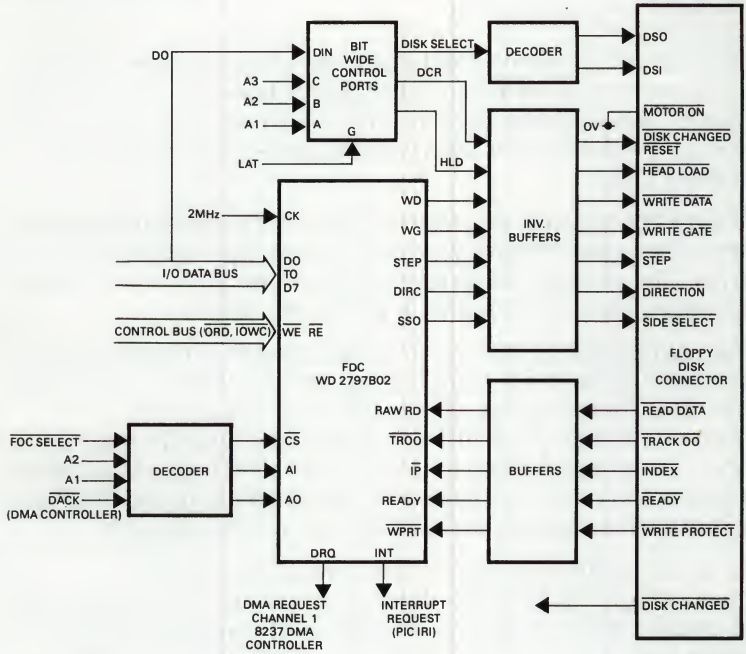


Figure 1. Floppy Disk Interface

FDC Pin Definition

WD	Write Data	DRQ	Data Request
WG	Write Gate	INTRQ	Interrupt Request
STEP	Step pulse output	CLK	Clock input
DIRC	Direction control	D0 to D7	Data bus
SSO	Side Select Output	$\overline{WE}$	Write Enable
RAW RD	Read Data	$\overline{RE}$	Read Enable
$\overline{TR00}$	Track 00	$\overline{CS}$	Chip Select
$\overline{IP}$	Index Pulse	A0,A1	Register selects
$\overline{READY}$	Ready input	$\overline{MR}$	Master Reset
$\overline{WPRT}$	Write Protect		

Four control lines for the disk drives are generated directly under software control, via the control port on the Interface Board. Two of these lines are the drive select lines (DS0 and DS1). The third is the Disk Change Reset line. The fourth is Head Load.

The motor control line (Motor On) is hard-wired to 0V, thus providing a permanently active control signal to the disk drive. The configuration of the drive is set so that the motor operates, only when a disk is within the drive.

The remaining functions for; controlling the movement of the read/write head, transferring data to and from the disks, and monitoring status signals from the drives are implemented by the FDC (apart from the Disk Changed input which is wired to an input port). A detailed description of all these connections and all other connections to the interface is provided in tabular form at the end of this chapter.

All disk transfer operations performed by the FDC (formatting, reading disk data, writing data onto disks), are controlled by the system software, and involve both the 8086 CPU and the DMA controller. The DMA controller is used to perform data transfers between system memory and the FDC on a byte-by-byte basis.

The actual sequence of events performed by the processors and the FDC during disk transfer operations is quite complex, involving the 8086 setting up the DMA controller and the generation of various disk drive control signals.

Due to this complexity, a simplified view of disk write, disk read and disk formatting operations is presented in the paragraphs below to illustrate some of the basic principles. A more detailed approach is taken in subsequent paragraphs.

Disk Write

Disk write operations are initiated by the CPU issuing a write command byte to the FDC over the I/O data bus. The FDC then searches for the required destination on the disk for the data (correct side, track and sector, with the correct ID field CRC bytes). When the required location is found, the FDC signifies to the DMA controller that a data byte is required by issuing a DMA request, via the DRQ output. The disk write cycle then proceeds as follows:

1. The DMA controller writes the first data byte into a holding register (termed the Data Register) within the FDC.
2. The FDC resets the DRQ output and transfers the contents of the Data Register to an encoding circuit, and signifies to the DMA controller that another data byte is required.
3. The encoding circuit converts the byte into MFM encoded data and then writes the data to the disk.
4. The DMA controller responds by writing the next data byte into the Data Register.
5. The process is then repeated.

After the last data byte in the sector is written onto the disk, a two-byte CRC is computed internally within the FDC, which is then also written onto the disk.

On completing a write cycle, the FDC generates an interrupt request on the INTRQ output.

The interrupt request indicates to the CPU the end of the command cycle (via the Interrupt Controller).

The CPU then can check a status register within the FDC to see whether any errors occurred during the write operation. Reading the Status Register or writing a new command to the FDC automatically resets the INTRQ output.

Disk Read

Disk read operations are initiated by the CPU issuing a read command byte to the FDC over the data bus. The FDC then searches the disk for the source of the data (correct side, track and sector, with correct ID CRC bytes). When the required location is found, the FDC begins the process of reading data from the disk.

The MFM encoded data is decoded, assembled into parallel data bytes, and transferred into the Data Register.

Every time a parallel data byte is transferred into the Data Register, the FDC signifies to the DMA controller that a data byte is available by issuing a DMA request, via the DRQ output.

Every time the DMA controller reads the assembled data byte, the FDC resets the DRQ output.

This process continues until the whole of the read operation is completed.

The FDC then generates an interrupt request on the INTRQ output, which indicates to the CPU the end of the command cycle. Any errors produced during the transfer can then be checked by the CPU analysing the FDC status register.

Disk Formatting

Disk formatting is a similar process to disk write operations but involves transferring both data and gap information (unintelligent information used to separate areas of data on the disk, see Figure 2) onto the disk, and is executed on a track-by-track basis.

The process is initiated by the CPU issuing a Write Track command to the FDC, via the peripherals data bus. The FDC responds by signifying to the DMA controller that a byte is required by issuing a DMA request, via the DRQ output.

The DMA controller then writes the first gap byte into the holding register (Data Register) within the FDC. The FDC resets the DRQ output and then waits for the Index Pulse (the marker for the start of a track) to be detected.

On detecting the Index Pulse, the FDC transfers the byte within the Data Register to an encoding circuit and issues another DMA request to the DMA controller. The DMA controller responds by writing the next byte to the Data Register and the process is repeated. The encoding circuit converts the byte into MFM data and writes the data onto the disk.

This process continues until the FDC detects the next Index Pulse, which causes the FDC to generate an interrupt request on the INTRQ output.

The interrupt request again has the same functions as described before; i.e. it indicates to the CPU the end of the command cycle. Any errors produced during the transfer can then be checked by the CPU analysing the FDC status register.

Resetting the INTRQ output is achieved by reading the Status Register or issuing a command to the FDC.

Read/Write Head Positioning

Prior to performing any of the disk transfer routines described above, the read/write head has to be positioned over the required track on the selected disk and the head load control signal issued to engage the read/write head. The head positioning operation is achieved under software control by issuing a command byte to the FDC.

The FDC responds to five different head positioning command bytes; each moves the head to a position specified by the command. The five commands are Restore, Seek, Step, Step-in and Step-out. The function of each command is detailed in the table below.

Head Positioning Commands

Command	Function
Restore	Positions the head over Track 0 on the Disk.
Seek	Positions the head to the track number specified in the Data Register.
Step	Moves the head to an adjacent track in the same direction as the previous head positioningcommand.
Step-in	Moves the head to the adjacent track in the direction away from Track 0.
Step-out	Moves the head to the adjacent track in the direction towards Track 0.

On completion of a head positioning command, the FDC generates an interrupt request on the INTRQ output to indicate to the CPU the end of the command cycle. Reading the Status Register or writing a new command to the FDC resets the INTRQ output.

An optional feature of each of the five commands is to automatically verify the track position of the head, by comparing the track number stored within an internal register (Track Register) with the track number contained in the ID fields on the disk.

The verification operation also checks the ID field for errors utilising the ID field CRC bytes. Failure to detect the same track number, or incorrect CRC detection, sets error status bits within the Status Register. Since this feature involves reading the disk ID field, the head has to be loaded prior to the command.

The signals supplied by the FDC to the disk drive, to position the read/write head are the STEP and DIRC (direction) outputs. A step pulse with a duration of 2 μs is issued to the drive every time the FDC wants to move the head by one track location.

The FDC determines the direction of movement implied by the command and sets the state of the direction output accordingly. (Logic high to move the head away from Track 0, logic low to move the head towards Track 0).

FDC Detail

General

The FDC can be divided from a descriptive point of view into three areas of circuitry; a processor interface, a disk drive controller, and a series of registers.

The processor interface handles the transfer of data, commands and status, between the internal registers and the system processors, and also generates the interrupt and DMA request signals.

The disk drive controller responds to commands from the CPU, providing the neccessary circuitry, and input and output lines for:

1. Controlling the positioning of the drive read/write head.
2. Transferring data to and from the disks.
3. Monitoring the disk drive status.

The internal registers provide the means of exchanging information (commands, status, positional data) between the disk drive controller and the processor interface.

Processor Interface

The connections to the processor interface are detailed in the table "System Connections" at the end of the chapter. The majority of the circuitry is a straightforward interface between the 8-bit bi-directional data bus and the series of addressable registers located within the FDC.

The system software views the registers as a series of peripheral ports located in the system input/output space. The port address locations, as defined by the FDC select and the system address bus connections, are detailed below. The Data, Sector and Track Registers can be both written to and read from. The Command Register can only be written to, and the Status Register can only be read from.

Register	Address	Transfer
Command	00H	Write Only
Status	00H	Read only
Track	02H	Bi-directional
Sector	04H	Bi-directional
Data	06H	Bi-directional

The remaining circuitry of the processor interface consists of the control section, which produces the INTRQ and DRQ outputs.

DMA Requests

The DRQ output is activated (set to logic high), during data transfer operations to and from the disk, and follows the state of an associated control bit within the Status Register. During disk read operations, the DRQ output is set active when the FDC has data available from the disk within the Data Register. The output is reset to the inactive state when the byte is read by the DMA controller.

During data write and formatting operations, the DRQ output is set active when the FDC Data Register is empty, and the FDC requires another byte from the DMA controller. The output is reset when the Data Register is loaded with a new byte.

Interrupt Requests

The INTRQ output is activated (set to logic high) on the successful completion of disk read, write, or formatting operations. It is automatically reset following these operations, or on reading the Status Register or when a new command is issued to the Command Register.

The INTRQ output is also set to logic high, for a variety of other conditions (premature termination of a disk transfer command sequence due to an error condition, completion of a read/write head position command, etc.).

In all cases, INTRQ can be reset by either reading the Status Register or issuing a command to the Command Register. These other conditions are detailed in the following paragraphs.

The first operation performed by the FDC on receiving a disk read or disk write command is to check whether the disk is ready for a transfer operation by analysing the READY input from the disk drive. If the input is set to logic high, the command sequence is initiated. If the input is set to logic low, the command sequence is immediately aborted, the Not Ready bit set within the Status Register, and the INTRQ output activated.

The FDC also analyses the Write Protect input ($\overline{\text{WPRT}}$) from the disk drive on receiving a disk write or formatting command. If the $\overline{\text{WPRT}}$ input is activated (logic low, indicating that the disk is write protected), the write operation is terminated, INTRQ is activated and the Write Protect bit set within the Status Register.

Prior to writing data to or reading data from the disk, the FDC locates the correct sector for the transfer operation, by analysing the ID fields on the disk. Failure to detect an ID field with the correct track number, correct side number, correct sector number and correct CRC within five revolutions of the Disk, sets the Record Not Found (RNF) bit in the Status Register, activates the INTRQ output and terminates the transfer operation.

At the start of disk write and formatting operations, the FDC signifies to the DMA controller that the first byte is required, via the DRQ output. If during formatting, the DMA controller fails to supply the first byte before the FDC detects the Index Pulse, the Lost Data (LD) bit is set within the Status Register, INTRQ is activated and the formatting operation is terminated.

If during disk writes, the DMA controller fails to supply the first byte before the start of the data field, the same process occurs; the Lost Data bit is set, INTRQ is activated and the operation is terminated.

During disk read operations, the FDC checks the two-byte CRC code at the end of the sector data field to ensure the validity of the data. If a CRC error is present, the CRC error bit in the Status Register is set, INTRQ is activated and the read operation is terminated (even if the operation is a multiple sector read).

In addition to the transfer commands, the INTRQ output is also activated at the end of the command sequence for positioning the read/write head of the drives. The success of the operation is again signified by the status bits within the Status Register.

The first operation performed by the FDC on receiving the Restore command is a check of the Track 0 input ($\overline{\text{TR00}}$). If the $\overline{\text{TR00}}$ input is set low (indicating that the head is already positioned over the first track), and the verification option is not selected, the FDC clears the Track Register to zero and activates the INTRQ output.

If the $\overline{\text{TR00}}$ input is high, the FDC issues step pulses until the $\overline{\text{TR00}}$ input is set low, which then produces the same effect as described above, providing the verification option is not selected (i.e. INTRQ activated, Track Register cleared). If the TR00 input is not set low within 255 step pulses, the operation is aborted, INTRQ is activated, and the Seek Error bit within the Status Register is set.

If the verification option is selected with any of the head positioning commands, the FDC moves the head to the specified position and then reads the first encountered ID field on the disk. The track number from the ID field is compared with the track number stored in the Track Register. Providing the two numbers are identical and the ID field CRC bytes are correct, the track position is deemed true, and the INTRQ output is activated.

If the track numbers match, but the CRC is incorrect, the CRC error bit within the Status Register is set and the next encountered ID field is analysed. If the FDC fails to detect an ID field with a matching track number and correct CRC within 5 revolutions of the disk, the operation is aborted, INTRQ is activated, and the Seek Error bit within the Status Register is set.

The FDC can be also issued with a command (Force Interrupt), which sets the Status Register to monitor the state of the input status lines from the drive, and causes the INTRQ output to be activated for any of the conditions detailed below:

1. Immediately the command is received.
2. Every time an Index Pulse is detected.
3. On detecting a transition on the Ready input.

Disk Drive Control

The disk drive controller circuitry acts as the interface between the disk drives and the other areas of circuitry within the FDC and consists of the disk data encoding and decoding sections, the head positioning control section and a status monitoring section.

Connections to and from the FDC disk controller circuitry are detailed in the table "Disk Drive Connections" at the end of the chapter.

At the start of a data transfer operation to the disk (write operation), the Write Gate output is activated and the DMA controller begins the process of transferring data bytes to the Data Register in parallel format, under DMA control.

The FDC transfers each byte from the Data Register to the encoding circuit. The encoding circuit converts the bytes into MFM double density encoded data, which is then supplied to the disk via the Write Data output.

When writing to tracks with a track number greater than 43, the encoding circuit provides automatic write precompensation. The precompensation value is set by a potentiometer (VR1), located on the Interface board.

The decoding section of the disk controller circuitry decodes the MFM encoded data from the Raw Read input, during transfer operations from the disk. The decoder is a phase-locked loop data separator circuit based around an internal voltage controlled oscillator (VCO) and phase detector.

The centre frequency of the VCO is set by a variable capacitor (VC1), located on the Interface Board. A second variable control VR2 sets the read window pulse width.

The head positioning control section responds to the head positioning commands supplied to the Command Register and controls the STEP and Direction (DIRC) outputs.

The rate at which the 2 μs step pulses are issued to the drive, to move the head from track-to-track, is specified by the command. Issuing a step pulse to the drive to move the head towards Track 0, automatically decrements the Track Register. Issuing a step pulse to the drive to move the head away from Track 0, automatically increments the Track Register.

The status monitoring section monitors the logic state on the four inputs from the disk drive, READY, $\overline{TR00}$, Index Pulse ($\overline{IP}$) and Write Protect ($\overline{WPRT}$). All the four inputs can be monitored by issuing any of the head positioning commands or the Force Interrupt command, and then examining the contents of the Status Register.

The effect of the status inputs on the operation of the FDC is dependent on the command issued to the Command Register and the logic state of the input line.

Command Register

The 8-bit Command Register holds the command supplied from the CPU which determines the type of operation carried out by the FDC, and is located at address location 00H in the I/O space. The register can only be written to with one of eleven predefined command words. A command currently in progress is indicated by an associated control bit within the Status Register.

The eleven commands can be divided into four different categories as detailed below. A detailed description of each command is provided in the Programming Considerations section.

Type 1. Read/write head positioning commands: Restore, Seek, Step, Step-in, Step-out.

Type 2. Data transfer commands: Read Sector, Write Sector.

Type 3. Format code transfer commands: Read Address, Read Track, Write Track.

Type 4. Interrupt command: Force Interrupt.

Status Register

The 8-bit Status Register holds status information, which is dependent on the command operation performed by the FDC. The register is located at address 00H in the system I/O space, and can only be read from. Some of the bits within the register signify the state of the control inputs from the disk drive, whilst others indicate the status of the command operation.

Track Register

The 8-bit Track Register indicates the track number of the position of the drive read/write head and is located at address location 02H in the system I/O space. The register can be written to and read from. The FDC updates the track register during head positioning command operations, every time the drive head is moved to an adjacent track.

Sector Register

The 8-bit Sector Register is used to store a sector number, which indicates to the FDC the desired location on the disk for a transfer operation. The register is located at address location 04H in the system input/output space and can be written to and read from. During multiple sector transfer operations, the sector register is updated by the FDC.

Data Register

The 8-bit Data Register is the holding register during transfer operations to and from the disk, and is located at address location 06H in the system I/O space. The register performs a different function during the Seek Command, when the register is programmed with the desired track location.

Programming Considerations

Disk Drive Selection

Selecting the integral disk drive for operation is achieved by writing data to a bit wide port mapped into the system I/O space at address location 30H. The port is connected to the LSB data line D0. Setting the port output to logic high, selects the integral disk drive for operation (i.e. writing FFH to I/O address 30H).

The Sony MicroFloppy Disk drive is selected by setting the logic states on the two drive select outputs (DS0 and DS1) to match a switch setting located on the drive.

Writing FFH to I/O address 30H matches the select state of the drive when it is configured as Drive 2. This the normal configuration switch setting of the integral disk drive. (The drive can be designated as Drive 1, 2, 3 or 4 according to the position of the switch - in theory allowing four drives to be integrated into a single system).

Writing 00H to I/O address 30H deselects drive 2 and matches the select state of a drive configured as Drive 3. This extra select state is provided to cater for any future upgrade of the Portable into a dual disk drive system, using an external MicroFloppy disk drive.

The two drive select outputs lines (DS0 and DS1) form a two-bit code which can be encoded into one of two different values only since DS0 is always the inverse of DS1. Writing FFH to 30H sets DS0 to logic high and sets DS1 to logic low. Writing 00H to 30H produces the inverse condition on DS0 and DS1.

Disk Change/Disk Change Reset

The Sony double-sided disk drive has a facility to allow the programmer to detect when a disk has been changed. This is provided by the input line "Disk Changed" from the disk drive. Disk Changed is set active (logic low) by the action of a disk being ejected from the disk drive. It remains in this state until the programmer resets the line using the Disk Changed Reset output.

The input line Disk Changed is wired to bit 5 (D5) of the octal buffer located at 24H in the system I/O space. Reading this bit allows the programmer to check the status of the input line. The data on D5 from this port mirrors the state of the input line (i.e. The bit is set to logic low when the disk is changed).

To reset the Disk Changed input line to the inactive state (logic high) requires the programmer to generate a 300 μs negative pulse on the Disk Changed Reset line. This is achieved by writing data to the I/O control port located at address 38H. This is a bit wide port formed by a bit addressable latch which uses the LSB data line D0 only.

The pulse is generated by simply writing FFH to I/O address 38H and after a delay of 300 μs, writing 00H.

Head Loading

Loading the read/write head of the selected disk drive is achieved by writing data to another bit wide port. This is mapped at 32H in the system I/O space. and is also connected to the LSB data line, D0 of the data bus.

The time taken for the Sony double-sided disk drive to engage the read/write head, after the head load signal is set active, is of the order of 60 ms.

A 60 ms delay should therefore be implemented after the head load signal is set active, to ensure the head is engaged before performing disk data transfer operations. All command operations can be performed with the head loaded without any detrimental effect to the disk.

Head Load Indicator

The BIOS is also able to provide an indication to the user that the head is loaded using the DISK indicator on the front panel of the Portable. This is controlled by writing to the LED section of the control port mapped at 02EH in the system I/O space.

This is a word wide (16-bit) port which also controls numerous other functions within the Portable. (As this port is write only, the BIOS keeps a copy of this port in RAM to allow individual bits within the port to be easily modified without adversely affecting the status of any of the other bits. The copy is located at the address given by adding an offset of 08H to the double word pointer stored in the RAM address location 722H.

The DISK LED is controlled by writing to bit 12 of the port. Writing a logic low to this bit position switches the LED on. Conversely, writing a logic high switches the LED off.

Head Positioning

Positioning the read/write head of the selected disk drive is achieved by issuing one of the five Type 1 commands to the Command Register of the FDC. Termination of a command (successful or otherwise) is signified by an interrupt request to the Interrupt Controller (PIC). The format of each of the head positioning commands is detailed below.

Type 1 Commands

D7							D0	Command
0	0	0	0	1	V	r1	r0	Restore
0	0	0	1	1	V	r1	r0	Seek
0	0	1	T	1	V	r1	r0	Step
0	1	0	T	1	V	r1	r0	Step-in
0	1	1	T	1	V	r1	r0	Step-out

T = Track Update Flag
V = Verify Flag
r1, r0 = Stepping motor rate

Each of the commands contains a Verify Flag bit and stepping motor rate bits. The stepping rate bits have to be set according to the track-to-track access time of the disk drive. The combination of bits allow four different stepping motor rates to be selected as detailed below. The track-to-track access time of the Sony double-sided disk drive is 12 ms.

r1	r0	Rate
0	0	3 ms
0	1	6 ms
1	0	10 ms
1	1	15 ms

The Verify Flag bit determines whether the FDC verifies the track position after positioning the head, by reading the sector ID fields on the disk. Verification is performed when the Verify Flag bit is set to logic high, and requires the head to be loaded prior to the command.

The verification operation checks the track number in the sector ID field with the number contained in the Track Register, and also checks the ID field CRC character.

Failure to match the track number or failure to find a matching track with a valid CRC, within five revolutions of the disk, causes the Seek Error bit to be set in the Status Register. If the verification operation detects a CRC error within any of the ID fields checked, the CRC error bit is set within the Status Register.

The Step commands contain a Track Update Flag, which determines whether the Track Register is updated every time the head moves to an adjacent track. The Track Register is updated if the Track Update Flag is set to logic high.

Restore Command

The Restore command is used to position the head over Track 0 of the disk. The FDC will issue up to 255 step pulses at the rate specified by the stepping rate bits, in an attempt to locate the first track on the disk.

Failure to locate Track 0 (by monitoring the state of the $\overline{\text{TR00}}$ input) within 255 step pulses, causes the command to terminate, and the Seek Error bit to be set within the Status Register. If the Verify Flag bit is set, verification of the track position is carried out as detailed above.

Seek Command

Prior to issuing the Seek command, the Data Register has to be loaded with the desired track number, and the Track Register is presumed to contain the current position of the head. On receiving the command, the FDC sets the DIRC output to move the head in the direction of the desired track.

Step pulses are then issued at the rate specified by the stepping rate bits, and the Track Register updated on each pulse, until the number in the Track Register coincides with the number in the Data Register. If the Verify Flag is set, verification of the track position is carried out.

Step Command

Issuing a Step command moves the head one track location in the direction specified by the previous command. If the Track Update Flag is set, the Track Register is updated following the issue of the Step pulse.

If the Verify Flag is set, verification of the track position is carried out, after a delay determined by the stepping rate bits (i.e. A delay of 12 ms for the Sony double-sided drive).

Step-in Command

Issuing the Step-in command moves the head one track location away from Track 0. If the Track Update Flag is set, the Track Register is incremented following the issue of the Step pulse.

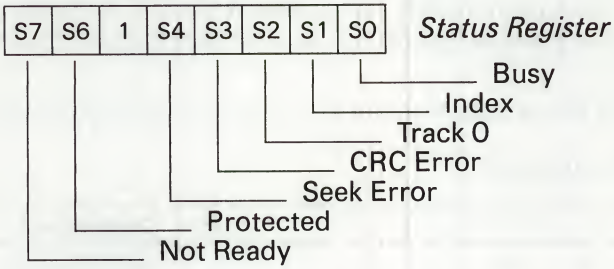
If the Verify Flag is set, verification of the track position is carried out, after a delay determined by the stepping rate bits.

Step-out Command

Issuing the Step-out command moves the head one track location towards Track 0. If the Track Update Flag is set, the Track Register is decremented following the issue of the Step pulse. The Verify Flag functions in the same manner as described above for the Step and Step-in commands.

Status Register

The format of the Status Register following a head positioning command is as detailed below. On issuing a command to the Command Register, the FDC resets the Status Register to monitor certain status conditions implied by the new command. Due to internal timing delays, the Status Register does not contain valid status information, until 14 μs after the new command is issued.



Busy	A logic high on Busy indicates that the command is in progress; logic low indicates that the command sequence is complete.
Index	Inverted copy of the Index Pulse input from the disk drive. Logic low when the pulse occurs.
Track 0	Inverted copy of the $\overline{\text{TR00}}$ input from the drive. Logic high indicates that the head is positioned over Track 0.
CRC Error	Logic high indicates a CRC error was detected within a sector ID field during verification of the track.
Seek Error	Logic high indicates a matching track number was not located within a sector ID field during the verification operation. The bit is also set high if the FDC fails to detect a logic low on the $\overline{\text{TR00}}$ input following a Restore command.
Protected	Inverted copy of the $\overline{\text{WPRT}}$ input from the drive. Logic high indicates that the selected disk drive contains a write protected disk.
Not Ready	Inverted copy of the READY input from the drive. Logic high indicates that the drive is not ready for a data transfer operation.

Data Transfers

Transfer of data to and from the disk is controlled by the two Type 2 commands, Write Sector and Read Sector. Prior to issuing the command to the Command Register, the Sector Register must be loaded with the desired sector number, to determine the source/destination of the data.

Termination of the command (successful or otherwise) is signified by an interrupt request to the PIC. Both command operations are prematurely terminated if the disk drive is not ready for the transfer operation, as indicated by the READY input.

The format of the data transfer commands is detailed below.

Type 2 Commands

D7							D0	Command
1	0	0	m	1	1	U	0	Read Sector
1	0	1	m	1	1	U	0	Write Sector

m = Multiple Record Flag
U = Update SSO

Both commands contain a Multiple Record Flag bit and an Update SSO bit. The Update SSO bit is used to select the disk side for the data transfer operations and affects the logic state of the Side Select Output (SSO), supplied to the disk drive. When U is set to logic low, the SSO is updated to logic low (side 0). When U is set to logic high, SSO is updated to logic high (side 1).

The Multiple Record Flag bit selects whether data transfers are from/to a single sector or multiple sectors within a track. Single sector transfers are initiated when the 'm' Flag is set to logic low, multiple sector transfers when the 'm' Flag is set to logic high.

Write Sector

On receipt of the Write Sector command, the FDC begins the process of searching the sector ID fields of the track for the desired destination for the data.

When an ID field is found with the correct track number (as specified by the Track Register), the correct side number (as specified by the U bit in the command), the correct sector number (as specified by the Sector Register), and correct CRC character; the FDC generates a DMA request via the DRQ output, to inform the DMA controller to write the first data byte into the Data Register.

If an ID field containing the correct information is not found within five revolutions of the disk, the command is aborted and the Record Not Found bit in the Status Register set.

If any of the ID fields encountered contain an incorrect CRC character, this is also recorded in the Status Register.

On receipt of the first data byte, the FDC activates the Write Gate output and, on detecting the start of the data field, writes the data byte to the disk. The process then continues with the FDC generating DMA requests every time a new byte of data is required.

After the 512th data byte is written to the disk, a two-byte CRC character is automatically generated and written onto the disk. If the data written to the sector only fills a part of the data field, the remaining area should be programmed with a series of zeroes, to complete the whole data field.

If a single sector write operation was specified by the Write Sector command, the Write Gate output is then deactivated and the command operation terminated.

If the Write Sector command specified a multiple sector transfer, the Sector Register is incremented and the process repeated, starting from the verification operation on the next ID field.

The multiple sector transfer operation continues until terminated either by issuing a Force Interrupt command, or after the sector number is incremented to a value exceeding the number of sectors on the track. In the latter case, the FDC automatically terminates the command after five revolutions of the disk, since the verification of the ID field will not be able to locate a matching sector number.

Due to the unpredictability of the FDC terminating the transfer in a "clean" manner, the use of multiple sector transfer operations as dictated by the FDC are of limited value. For this reason and because the majority of applications only generally request single sector transfers, the BIOS only supports single sector operations.

Fast "multiple" single sector transfers are supported by using a sector interleave factor of 2 (Interleaving is the term given to altering the physical order of sectors within a track to improve the access time when transferring more than one sector).

Failure of the DMA controller to write the first data byte to the Data Register before the arrival of the sector data field causes the FDC to; set the Lost Data bit in the Status Register, activate the INTRQ output, and abort the command.

Failure of the DMA controller to supply a data byte to the Data Register on receiving a DMA request after the first byte (i.e. within 13.5 μs), causes the FDC to write a byte of zeroes onto the disk and also set the Lost Data bit, but does not terminate the command sequence.

Read Sector

On receipt of the Read Sector command, the FDC begins the process of searching the sector ID fields of the track for the desired source of data.

When an ID field is found with the correct track number (as specified by the Track Register), the correct side number (as specified by the U bit in the command), the correct sector number (as specified by the Sector Register), and the correct CRC character; the FDC reads the data bytes from the following data field, and informs the DMA controller, via the DRQ output, every time a data byte is stored in the Data Register.

If an ID field containing the correct information is not found within five revolutions of the disk, the command is aborted and the Record Not Found bit in the Status Register set.

If any of the ID fields encountered contain an incorrect CRC character, this is also recorded in the Status Register.

Failure of the DMA controller to read the contents of the Data Register before it is overwritten with the next byte of data from the disk causes the FDC to set the Lost Data bit in the Status Register.

If a single sector read operation was specified by the Read Sector command, the sequence terminates with the FDC testing the CRC character at the end of the data field. If the CRC character is incorrect, the CRC Error bit is set within the Status Register.

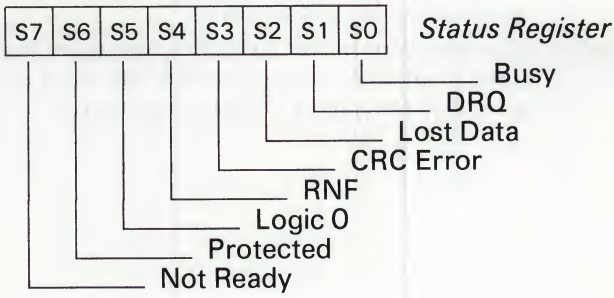
If the Read sector command specified a multiple sector transfer, the Sector Register is incremented and the process repeated for the next sector, providing the CRC character tested at the end of the previous data field was true. If the CRC character was incorrect, the multiple sector routine is prematurely aborted.

If no CRC Errors are detected, the multiple sector operation continues until terminated either by issuing a Force Interrupt command, or after the sector number is incremented to a value exceeding the number of sectors on the track. In the latter case, the FDC automatically terminates the command after five revolutions of the disk, since the verification of the ID field will not be able to locate a matching sector number.

The BIOS does not support multiple sector transfers (see Write Sector section above).

Status Register

The format of the Status Register following a data transfer command is as detailed below. On issuing a command to the Command Register, the FDC resets the Status Register to monitor certain status conditions implied by the new command. Due to internal timing delays, the Status Register does not contain valid status information, until 14 μs after the new command is issued.



Busy	A logic high on Busy indicates that the command is in progress; logic low indicates that the command sequence is complete.
DRQ	A copy of the DRQ output to the DMA controller. A logic high indicates that the FDC has data available during a read operation, and requires data during a write operation.
Lost Data	Logic high indicates that the DMA controller did not respond to the DRQ output in time, and as a result; valid data was not written onto the disk during a write operation, valid data was not read from the disk during a read operation.
CRC Error	During the write operation, a logic high indicates a CRC error was detected within one or more sector ID fields. During the read operation, a logic high indicates a CRC error was detected either within one or more ID fields, or a CRC error detected at the end of a data field.
RNF	Record Not Found. A logic high indicates that the transfer operation was unable to locate an ID field containing the correct side, track and sector number, with a valid CRC character.
Protected	Inverted copy of the $\overline{\text{WPRT}}$ input from the drive, during write operations. Logic high indicates that the selected disk drive contains a write protected disk. During read operations set to logic low.
Not Ready	Inverted copy of the READY input from the drive. Logic high indicates that the drive is not ready for a data transfer operation.

Formatting Commands

The formatting commands are the three Type 3 commands, Write Track, Read Track and Read Address. Write Track is the command for formatting disks. The two other commands enable the programmer to check the disk format.

Termination of the command is signified by an interrupt request to the Interrupt Controller. The command operations are terminated prematurely if the disk drive is not ready for the transfer operation as indicated by the READY input.

The format of these commands is detailed below.

Type 3 Commands

D7							D0	Command
1	1	0	0	0	1	U	0	Read Address
1	1	1	0	0	1	U	0	Read Track
1	1	1	1	0	1	U	0	Write Track

U = Update SSO

Each of the three commands contain an Update SSO bit. This bit is used to select the disk side for the formatting operations and affects the logic state of the Side Select Output (SSO) supplied to the disk drive. When U is set to logic low, SSO is updated to logic low (side 0). When U is set to logic high, SSO is updated to logic high (side 1).

Track Format

The track format used on the MicroFloppy disks is illustrated at the end of the chapter.

Read Address

On receipt of the Read Address command, the FDC searches the disk for an ID field. The first ID field encountered is then read from the disk. The FDC transfers the six ID field bytes, detailed below, from the disk to the Data Register one byte a time. Every time a data byte is stored in the Data Register, the FDC generates a DMA request to the DMA controller. The ID field track number is also transferred to the Sector Register.

Byte	Description
1	Track Number
2	Side Number
3	Sector Number
4	Sector Length (02H)
5	CRC 1
6	CRC 2

If the FDC fails to locate an ID field within five revolutions of the disk, the command is aborted and the Record Not Found bit in the Status Register set. If the ID field contains an incorrect CRC character, this fact is also signalled in the Status Register.

Failure of the DMA controller to read the contents of the Data Register, before the register is overwritten with the next byte of data from the disk, causes the FDC to set the Lost Data bit in the Status Register.

Read Track

The Read Track command allows a complete track of information (Gaps, Headers and Data bytes) to be read from the disk. On receipt of the Read Track command, the FDC monitors the logic state on the Index Pulse input. On detecting the leading edge of the pulse, the FDC transfers all the Gap, Header and Data bytes from the track into the Data Register on a byte-by-byte basis.

Every time a byte is transferred into the Data Register, the FDC informs the DMA controller that a byte is available, via the DMA request line. The command terminates after one full revolution of the disk, on detecting the Index Pulse again. No CRC error checking is carried out.

Every time the FDC detects the ID field and Data field address marks, it synchronises the internal decoding circuitry, to ensure that all information following each sector ID address mark is valid. Since the FDC might not be synchronised to the gap information previous to the ID address mark, this information is not guaranteed to be valid.

Failure of the DMA controller to read the contents of the Data Register, before the register is overwritten with the next byte of data from the disk, causes the FDC to set the Lost Data bit in the Status Register.

Write Track

The Write Track is the command used to format the tracks on the disk. All data and gap information is supplied to the disk by building a image of the track in memory as detailed on the next page, and transferring the format data to the Data Register under DMA control, using the DMA controller.

On receipt of the Write Track command, the FDC generates a DMA request via the DRQ output, to inform the DMA controller to write the first byte from the track image into the Data Register.

On detecting the leading edge of the Index Pulse, the FDC transfers the first byte to the disk and requests another byte from the DMA controller.

The process then continues with the DMA controller supplying the format bytes with the FDC generating DMA requests every time a new byte is required.

The command sequence terminates after one full revolution of the disk, on detecting the leading edge of the next Index Pulse.

Track Image

Number of bytes required	Byte Value (Hex)	Description	
80	4E	Gap	
12	00	Sync	
3	F6	Control Bytes	*PREAMBLE*
1	FC	Index Mark	
50	4E	Gap	
12	00	Sync	
3	F5	Control Bytes	
1	FE	ID Address Mark	
1	00 to 50*	Track Number	
1	00 or 01	Side Number	
1	01 to 09	Sector Number	*SECTOR*
1	02	Sector Length	(Repeated 9
1	F7	CRC character command	times
22	4E	Gap	changing
12	00	Sync	the Sector
3	F5	Control Bytes	Number each
1	FB	Data Address Mark	time)
512	00	Data Bytes	
1	F7	CRC character command	
84	4E	Gap	
598**	4E	Gap	*POSTAMBLE*

* For 80 track disks.

** Nominal figure, write operation continues until terminated by the INTRQ output, on detecting the Index Pulse.

Some of the bytes supplied to the Data Register are not written onto the disk, but act as control bytes to the FDC. These are the three bytes F5H, F6H and F7H.

F5H commands the FDC to perform two functions; write a unique byte pattern corresponding to A1H with a missing clock transition onto the disk and initialize the CRC generator circuitry. F6H commands the FDC to write a unique pattern corresponding to C2H with a missing clock transition onto the disk.

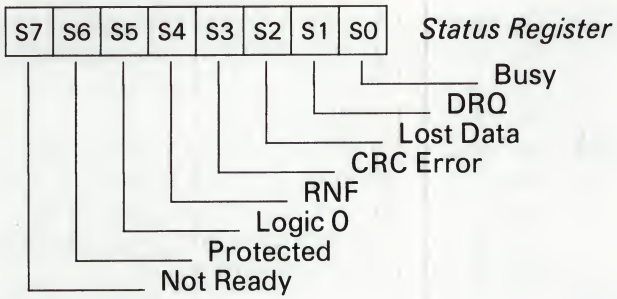
These patterns enable the FDC to interpret the byte following the three byte pattern as an address mark, during decoding operations. The byte F7H commands the FDC to write the computed two-byte CRC character onto the disk.

Failure of the DMA controller to write the first byte to the Data Register before the arrival of the leading edge of the Index Pulse causes the FDC to; set the Lost Data bit in the Status Register, activate the INTRQ output, and abort the command.

Failure of the DMA controller to supply a data byte to the Data Register on receiving a DMA request after the first byte, causes the FDC to write a byte of zeroes onto the disk and also set the Lost Data bit, but does not terminate the command sequence.

Status Register

The format of the Status Register following a formatting command is as detailed below. On issuing a command to the Command Register, the FDC resets the Status Register to monitor certain status conditions implied by the new command. Due to internal timing delays, the Status Register does not contain valid status information, until 14 μs after the new command is issued.

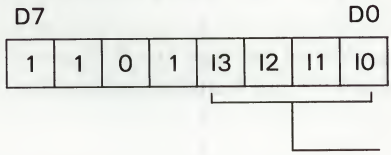


Busy	A logic high on Busy indicates that the command is in progress; logic low indicates that the command sequence is complete.
DRQ	A copy of the DRQ output to the DMA controller. A logic high indicates that the FDC has data available during the Read Track/Read Address operations, and requires data during a Write Track operation.
Lost Data	Logic high indicates that the DMA controller did not respond to the DRQ output in time, and as a result; valid data was not written onto the disk during a Write Track operation/valid data was not read from the disk during a read operation.
CRC Error	During the Read Address operation, a logic high indicates a CRC error was detected within the sector ID field. During the Read Track and Write Track operations set to logic low.
RNF	Record Not Found. A logic high indicates that the Read Address operation was unable to locate an ID field. During Read Track and Write Track set to logic low.
Protected	Inverted copy of the $\overline{\text{WPRT}}$ input from the drive, during the Write Track operation. Logic high indicates that the selected disk drive contains a write protected disk. During read operations set to logic low.
Not Ready	Inverted copy of the READY input from the drive. Logic high indicates that the drive is not ready for a data transfer operation.

Force Interrupt Command

The Force Interrupt command is a Type 4 command and is used to force the FDC to generate an interrupt via the INTRQ output on detecting a certain condition. The conditions are defined by setting certain bits within the command byte as detailed below.

Type 4 Command



Interrupt Condition Flags

I3	I2	I1	I0	Interrupt Condition
x	x	x	1	Not Ready to Ready Transition
x	x	1	x	Ready to Not Ready Transition
x	1	x	x	Every Index Pulse
1	x	x	x	Immediate Interrupt
0	0	0	0	No Interrupt

More than one condition can be set at any time by the appropriate combination of Interrupt Condition Flags.

If any of the Force Interrupt commands are issued to the Command Register when a command is already in operation, the current command is terminated and the Busy bit in the Status Register reset. All the other bits in the Status Register are left unchanged.

If the command is issued when there is no other command currently in operation, the Status Register is set to monitor the same conditions as a Type 1 command, as previously described.

The No Interrupt command functions in an identical manner to the other Force Interrupt commands, as described in the paragraph above, but does not produce an interrupt request on INTRQ.

The INTRQ output is reset following any of the Force Interrupt commands, by issuing any new command to the Command Register or by reading the Status Register. The only method available to reset the INTRQ output after issuing the Immediate Interrupt command, is to issue the No Interrupt command.

After issuing a Force Interrupt command to the Command Register:

1. Another command should not be issued for at least 8 μs, to allow the FDC to process the interrupt command.
2. The Status Register should not be read for 14 μs to ensure the register contains valid status information.

Interface connection Detail

System Connections

D0 to D7	Data bus. Used to transfer data, commands and status information between the CPU/DMA controller and the FDC.
$\overline{WE}$	Write Enable. Write control input connected to the $\overline{IOWC}$ control line of the system control bus. Active state, logic low. Used in conjunction with $\overline{CS}$ and the register select inputs (A0, A1) to transfer data and commands from the data bus to the FDC.
$\overline{RE}$	Read Enable. Read control input connected to the $\overline{IORC}$ control line of the system control bus. Active state, logic low. Used in conjunction with $\overline{CS}$ and the register select inputs (A0, A1) to transfer data and status information onto the data bus from the FDC.
$\overline{CS}$	Chip Select. Address input. Active state, logic low. When active, indicates that the FDC is selected for a data/command transfer operation.
A0, A1	Register select lines. Inputs connected to A1 and A2 of the system address bus and DACK of the DMA Controller via a pair of OR gates. Used to select internal registers within the FDC for data/command/status transfers, via the data bus as detailed below.

A1	A0	
0	0	Status/Command registers
0	1	Track register
1	0	Sector register
1	1	Data register

INTRQ	Interrupt Request. Output connected to an interrupt request line of the Interrupt Controller (PIC) on the CPU and Display Board. Active state, logic high. When active, signifies that the FDC has terminated a command operation.
DRQ	Data Request. Output connected to the DMA request line of channel 1 of the DMA controller (DREQ1). Active state logic high. When active, signifies to the DMA controller that the FDC is ready for a DMA transfer operation.
$\overline{\text{MR}}$	Master Reset. Input connected to the System Reset control line. Active state, logic low. When active, causes the FDC to reset its internal registers.
CK	Clock input. 2 MHz clock with a 50% duty cycle for internal timing within the FDC.

Disk Drive Connections

$\overline{DS0}$	Drive Select 0. Control signal driven by a single-bit control port mapped in the system I/O space. Active state, logic low. When active, selects the disk drive configured as drive 2.
$\overline{DS1}$	Drive Select 1. Control signal driven by a single-bit control port mapped in the system I/O space. Active state, logic low. When active, selects the disk drive configured as drive 3.
HLD	Head Load. Control signal driven by a single-bit control port mapped in the system I/O space. Active state, logic high. When active, engages the read/write head of the selected disk drive against the disk.
SSO	Side Select Output. Control signal from the FDC used to select side 0 or side 1 of double-sided disks. Follows the state of an associated control bit within an internal register. A logic high on SSO selects side 0 and a logic low, side 1.
STEP	Step Pulse. Pulsed output generated by the FDC for positioning the disk drive read/write head. Each positive going pulse moves the head to an adjacent track location in the direction determined by the DIRC output.
DIRC	Direction Control. Control signal generated by the FDC to determine the direction of movement of the disk drive read/write head. When DIRC is set to logic high, each step pulse causes the head to step in one track (away from track 0). When DIRC is set to logic low, each step pulse causes the head to step out one track (towards track 0).
WG	Write Gate. Control signal generated by the FDC. Active state, logic high. When active, enables current to flow into the disk drive read/write head.
WD	Write Data. Output for writing MFM encoded data to the disk drive.

RAW RD	Raw Read. Input to the FDC for MFM encoded data from the disk drive.
READY	Input to the FDC from the disk drive. When set to logic high, indicates that the selected disk is ready for data transfer operations. When set to logic low, indicates that the selected disk is not available. In this condition attempted data transfer operations between the FDC and the disk drive are inhibited, and cause an active interrupt request to be generated on INTRQ. READY also sets an associated control bit within an internal register according to the logic state on the control line.
$\overline{\text{TR00}}$	Track 00. Control input from the disk drive. When $\overline{\text{TR00}}$ is set to logic low, indicates to the FDC that the read/write head of the selected drive is positioned over track 0 of the disk.
$\overline{\text{IP}}$	Index Pulse. Control input from the disk drive. A negative going pulse generated every revolution of the disk, which informs the FDC of the start of the first sector of each track.
$\overline{\text{WPRT}}$	Write Protect. Control input from the disk drive. When $\overline{\text{WPRT}}$ is set to logic low, any attempted write operation to the selected drive is inhibited. $\overline{\text{WPRT}}$ also sets an associated control bit within an internal register according to the logic state in the control line.

Track Format

Data is recorded on a disk in concentric circles, known as tracks. When a disk is inserted into the disk drive, the Auto Shutter is opened to allow the read/write head of the disk drive access to the disk surface.

When the shutter is open and the head loaded, the head makes physical contact with the radial slot of magnetic material exposed. Information on a track is read and written serially as the disk rotates within its protective plastic shell.

Each track of the disk is divided into nine sectors by software formatting (soft sectoring). The sectors are recorded onto each track of the disk by issuing a format command to the FDC and then writing all the bytes onto the disk as illustrated in Figure 2.

The start of each track is marked by a single index pulse, which is generated by the disk drive every revolution of the disk.

Each sector has an identification (ID) field and a data field, separated by gaps of unintelligent information.

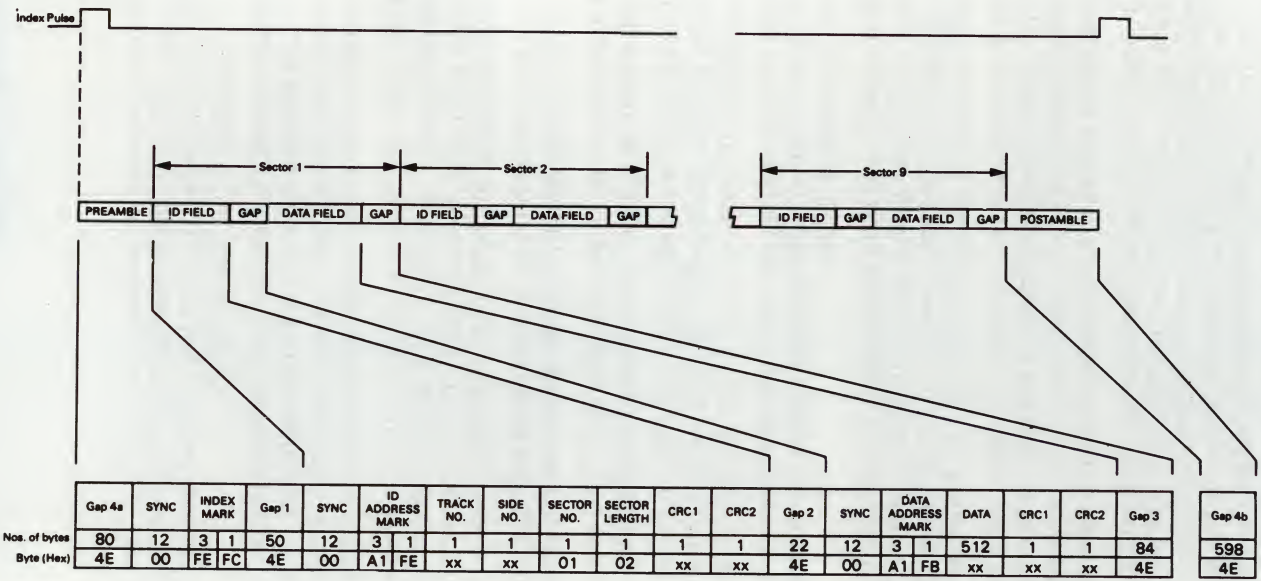
The gaps are required to allow the FDC to process information from the disk during disk reads, and also to take into account variations in drives, such as motor speed variations.

The ID field defines the data field that follows, by specifying its track number, side number, sector number and length of the data field in bytes.

Both the ID and data fields begin with an address mark and end with a two-byte cyclic redundancy check (CRC) character for detecting errors in the previous field. Different address marks are used to distinguish between the two fields.

All the serial data on the track (ID fields, data fields and gap information) are recorded on the disk by Modified Frequency Modulation (MFM). In MFM encoding, both serial data bits and clock pulses are interleaved into the data stream.

In order to distinguish the address marks on the disk from data bytes which may be identical, the address marks are recorded with missing clock pulses in predefined locations. The address marks are the only bytes on the disk with missing clock bits, and are produced by the FDC during formatting. These are then used by the FDC for recognition and synchronisation during disk accesses.



	Gap 4a	SYNC	INDEX MARK		Gap 1	SYNC	ID ADDRESS MARK		TRACK NO.	SIDE NO.	SECTOR NO.	SECTOR LENGTH	CRC1	CRC2	Gap 2	SYNC	DATA ADDRESS MARK		DATA	CRC1	CRC2	Gap 3	Gap 4b
Nos. of bytes	80	12	3	1	50	12	3	1	1	1	1	1	1	1	22	12	3	1	512	1	1	84	598
Byte (Hex)	4E	00	FE	FC	4E	00	A1	FE	XX	XX	01	02	XX	XX	4E	00	A1	FB	XX	XX	XX	4E	4E

Figure 2. Track Format

Serial Interface 2.7

Contents

Illustrations

Introduction

The Serial Interface is located on the Interface Board and consists of the elements of circuitry as illustrated on the block diagram Figure 1. The interface provides two separate serial channels; one channel for receiving data from the Infra-red (IR) Keyboard or Mouse, the second channel for communication between the Portable and external equipment via an RS232C link.

The receive channel for the Keyboard/Mouse data is programmed to operate in synchronous mode (Monosync) at a fixed data rate. It is supplied with keyboard/mouse data via the IR receiver board which decodes the incoming data and converts it into an acceptable serial waveform. (Details of the IR Receiver Board are discussed in the chapter headed Systems Unit).

The RS232C channel can be programmed to operate in either asynchronous or synchronous modes, with transmit and receive baud rates determined either via the Programmable Interval Timer (PIT), or via the external data communications equipment.

The RS232C interface is able to support:

1. Asynchronous communications with 5, 6, 7 or 8 bits per character and various choices of stop bits and parity sense.
2. Bit oriented synchronous communications, such as SDLC and HDLC.
3. Byte oriented synchronous communications, such as Monosync and Bisync.

Details

General

The Serial Interface consists of; a serial input/output controller, a data selector, a number of line drivers/receivers, a signal conditioning circuit which processes the incoming keyboard/mouse data and a connector (25 pin D-type female connector for the RS232C channel).

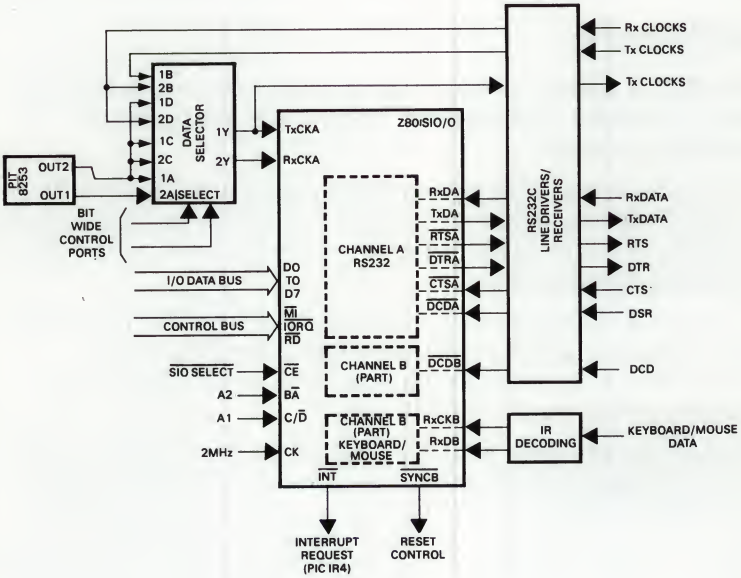


Figure 1. Serial Interface

The major circuit element of the Serial Interface is the serial input/output controller which is a Zilog Z80 SIO/0. The SIO is a programmable device which incorporates all the circuitry for interfacing between the 8086 CPU and the two serial channels. A description of each connection to the SIO is provided at the end of the chapter.

The data selector allows the programmer to select the clock source which determines the transmit and receive baud rates for the RS232C channel.

Due to the relative complexity of the SIO compared with other programmable devices on the Interface Board, a brief overview of the device is presented in the following paragraphs.

Subsequent paragraphs describe the internal structure and the utilisation of the device within the system in more detail.

SIO Overview

The SIO is a peripheral device, which performs the function of an intelligent two-channel parallel to serial/serial to parallel converter.

The two channels (designated Channel A and Channel B) are totally independent of each other and can be programmed to operate in either asynchronous or synchronous serial data communications modes.

Channel B is used as a uni-directional channel for receiving synchronous data from either the keyboard or mouse. Channel A is available to the programmer for use as a full duplex, multi-mode RS232C link.

Both channels have to be initialised with the required serial data communications mode, prior to any transmission or reception. The programmer is then able to view the SIO as two simple parallel input/output ports for the majority of data transfer operations; one bi-directional port for Channel A, one (read only) input port for channel B (since Keyboard communications are uni-directional).

The SIO provides all the necessary facilities for organising data transfers over the two serial channels. These include:

1. Conversion of the parallel data into the correct serial data format for transmission over the selected serial channel. In asynchronous modes, this involves the automatic insertion of start, stop and parity bits, and assembling the parallel data into the correct number of serial bits per character.
2. Conversion of the input serial data from the serial channels into parallel form, and checking the incoming data for errors. In asynchronous modes, this involves stripping the start, stop and parity bits from the serial data and checking for parity and framing errors.

When operating in synchronous modes, similar automatic facilities are available within the SIO, dependent on the selected mode.

In Bisync and Monosync, pre-programmed sync characters and CRC characters can be automatically inserted during transmissions. The SIO strips sync characters from received data, and also analyses the received data for errors, utilising the received CRC characters.

In SDLC and HDLC, the SIO performs the following functions:

1. Abort sequence generation and detection.
2. Zero insertion and deletion.
3. Flag insertion between messages.
4. Address field recognition.
5. I-field residue handling.
6. CRC generation and detection.

The baud rates for the two channels are determined by clock sources external to the SIO. In asynchronous modes, the clocks can be divided internally within the SIO under program control, prior to being used to set the baud rates. This facility is not available in synchronous modes.

Each channel also provides inputs and outputs which are specifically tailored to function as modem control lines for co-ordinating serial data transfer operations. The outputs are under direct control by the programmer.

The inputs can be monitored under software control by polling, or be programmed to generate an interrupt to the CPU. These control lines can also be redefined as general purpose input and output control signals.

The SIO can also be programmed to generate interrupts to signal to the processors the status of various transmit and receive conditions in the two communication channels. These include; receive data available, transmit data required, and the detection of various error conditions.

The interrupt structure of the SIO is such that on generating an interrupt to the CPU, it can also produce an interrupt vector internally which defines the cause of the interrupt. This can then be used by the CPU to point to an associated service routine. The vector is obtained from the SIO by reading a SIO register.

The SIO only generates an interrupt to the CPU when a channel requires a specific servicing routine to be carried out, (e.g. keyboard data available) thus leaving the CPU free to service the other devices on the board.

The interrupt structure of the SIO acts as an extension of the interrupt structure provided by the Interrupt Controller (PIC) on the CPU and Display Board.

The SIO interrupt is not wired directly to the CPU but is supplied via the PIC. The PIC provides a vector to the CPU to indicate the device generating the interrupt. If the device is the SIO, the actual process within the device requiring attention is then specified by the interrupt vector from the SIO.

SIO Architecture

Internally, the SIO consists of four areas of circuitry:

1. A Processor interface.
2. Internal interrupt control logic.
3. The two independent serial communication channels, channel A and channel B.

The processor interface handles all communications via the data bus between the CPU and a series of internal registers, associated with each of the two serial channels. These are of three different types; data registers, command registers and status registers.

Both channels have two data registers each which are accessed by the CPU; one handles the transmit data from the 8086 - transmit buffer, and one stores receive data from the serial channel - receive buffer.

The mode and operation of each channel is determined by the contents of the command registers (Write Registers). These are initialised with control words prior to any data transfer over the two serial channels, and may be modified as data transfer operations proceed.

Seven Write Registers are associated with each channel. An eighth Write Register, which is located in channel B is shared by both channels.

The information contained in the status registers (Read Registers) indicate the status within each channel. Two Read Registers are associated with each channel. A third register accessed via channel B is common to both channels.

The interrupt control logic section of the SIO assigns the priority to the various sources of interrupts, generates the interrupt output to the PIC and produces the interrupt vector.

The priority of the interrupts is fixed with channel A always assigned a higher priority than channel B. The order of priority assigned within each channel is receiver interrupts (highest priority), followed by transmitter interrupts, followed by external/status interrupts (lowest priority).

Separate transmit and receive data paths are provided within the two serial communications channels. The transmit data path of channel B is redundant, since the keyboard data flow is uni-directional, to the Systems Unit only.

Receive Path

The receiver ports are quadruply buffered by an input shift register and three storage registers. The shift register converts the incoming receive serial data into a parallel byte.

The three storage registers are configured in a FIFO (first in first out) arrangement. The first parallel data byte from the shift register is loaded into the bottom of the FIFO stack and then shifted through to the top at a rate, determined by an internal clock.

The register at the top of the stack is the receive buffer which acts as the storage buffer for the receive character until read by the processors. Reading the character in the receive buffer automatically transfers the next character (if any) to the top of the stack.

Error flags associated with each receive character are also buffered by a similar arrangement. These are loaded into the register of a parallel receive error FIFO stack at the same time as the character. Each time the receive character is shifted through the character FIFO stack, the error flags are moved accordingly. The top of the receive error FIFO stack is one of the Read Registers.

The contents of this Read Register reflect the status of the character stored in the receive buffer, but may also contain any receiver overrun and parity errors received from previous characters. These two error conditions remain within the status register, even when new character error flags are loaded, until cleared by an error reset command.

If the character FIFO stack is full (i.e. contains three characters) and the CPU fails to read the character within the receive buffer before another receive character is supplied to the stack, the last character placed in the stack is overwritten with the new character, and the receiver overrun flag is recorded in the corresponding register in the receive error FIFO stack.

The first two characters in the stack are never overwritten, even if more characters are received; the last character is continuously overwritten.

Error conditions stored in the FIFO stack can be programmed to generate an interrupt to the CPU, on being loaded into the Read Register at the top of the stack.

In the asynchronous mode, using 8 bits per character, the serial receive data is stripped of the start, stop and parity bits, prior to being supplied to the serial shift register.

For character lengths of less than 8 bits, the receiver circuits automatically insert logic 1's in the most significant places to assemble a byte of data, if a parity bit is not included with the character. If a parity bit is included, the parity bit is also assembled with the bits of the character, with any remaining bits set to logic 1.

For example, a 5-bit receive character with a parity bit is assembled in the receive buffer in the following format:

1	1	P	D4	D3	D2	D1	D0

D = 5-bit character.
P = Parity bit.

In the byte oriented synchronous modes, Monosync and Bisync, receive data is not transferred to the receive FIFO register stack until character synchronisation is established.

Detecting a byte of receive data which matches a sync character stored in a Write Register establishes synchronisation in Monosync. Detecting two consecutive bytes of receive data which match the sync characters stored in two of the Write Registers establishes synchronisation in Bisync.

Searching for the character sync is achieved by programming the channel to operate in the first phase of the synchronous reception process, termed the hunt phase. The second phase is the actual reception of data, where the receive data is automatically transferred to the FIFO stack after detection of the character sync.

Also included in the receive path in synchronous modes, is a CRC error detection circuit which sets an error flag according to the result of CRC comparisons.

The receiver circuits in the bit oriented synchronous modes operate in a similar manner to the byte oriented modes, operating with two separate phases, a hunt phase and a receive phase. Receive data is not transferred to the receive FIFO register stack until the hunt phase is sucessfully completed.

This requires detecting an opening flag sequence corresponding to a flag pattern, stored in one of the Write Registers. Reception is terminated on detecting the closing flag at the end of the message (EOM). This prevents any further data being supplied to the FIFO register stack. The receive data is also supplied to the CRC error detection circuits.

Transmit Path

The transmitter sections of the serial channels consist of an 8-bit data register (the transmit buffer), supplied with character data from the CPU, and a 20-bit transmit shift register. The shift register converts the parallel data in the transmit buffer into serial format and also performs a variety of other functions depending on the mode of operation.

In the asynchronous mode, the data from the transmit buffer is automatically formatted with start, stop and parity bits within the shift register, prior to transmission.

In Monosync and Bisync, the shift register is loaded with the sync characters stored in the Write Registers, at the beginning of the message, and then data from the transmit buffer as transmission proceeds. The shift register also supplies the serial data to the CRC generator, which produces the two byte CRC at the end of the message.

In the bit oriented modes, the shift register is loaded with the flags stored in the Write Register, at the beginning and end of the message. The shift register supplies the serial data to the CRC generator which generates the 2-byte CRC at the end of the data field, and to the transmit data output.

For character lengths of less than 8 bits, the characters sent to the transmit buffer have to be right justified, by the programmer. The logic state of the unused bits within each character byte (the MSB) are immaterial for character lengths of 6 and 7 bits, since the extra bits are automatically ignored by the SIO.

For a 5-bit character, the unused bits have to be programmed with logic 0's. Character lengths of less than 5 bits require a combination of logic 1's and logic 0's, to be inserted in the MSB as detailed in a later section.

Processor Interface

The connections to the SIO processor interface are detailed in the table "System Connections" at the end of the chapter. The interface handles the transfer of data, commands and status information (via the system data bus), between the CPU and the series of addressable registers within the SIO.

The system software views the SIO as four bi-directional peripheral ports, located in the system I/O space. The port addresses, as defined by the SIO select and the system address bus connections, are detailed below.

Address	Port
18H	Channel A Tx/Rx data
1AH	Channel A commands/status
1CH	Channel B Tx/Rx data
1EH	Channel B commands/status

Writing data to the I/O address location 18H loads data into the transmit buffer of channel A (RS232C transmit data); reading data from the same location accesses data stored in the receive buffer of channel A (RS232C receive data).

Reading data from I/O address location 1CH accesses data stored in the receive buffer of channel B (Keyboard/mouse data).

Writing data to the Write Registers (command registers) and reading data from the Read Registers (status registers) in the two channels, generally requires two data transfers.

The first transfer is a write to the commands/status address of the channel, and provides the SIO with a pointer to determine the register for the next read or write operation.

If the next operation is writing to the command/status address location, a command byte is loaded into the Write Register specified by the pointer. If the next operation is reading from the commands/status location, a status byte is accessed from the Read Register specified by the pointer.

Following the second transfer operation the pointer within the SIO is always cleared to zero.

Write Register Definition

Each channel contains seven Write Registers, which configure the SIO to match the desired mode and application. Five of these registers are command registers which define the basic mode of operation (e.g. asynchronous), and configuration of the channel within the mode (e.g. number of bits per character, etc.).

The two other registers are sync character registers. An eighth register, which is written to via channel B, is common to both channels and is programmed with the base address of the interrupt vector.

A summary of the functions of the Write Registers is provided in tabular format below. This is followed by a more detailed description of the commands for each individual register.

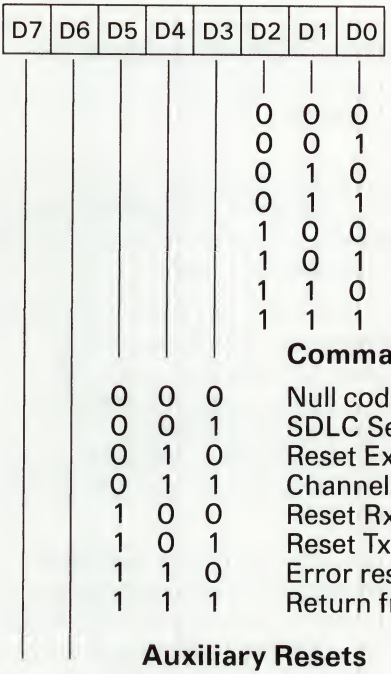
The only constraint on the order of programming the registers is that initialising each channel of the SIO following a hardware system reset or a software channel reset, requires the parameters of Write Register 4 to be issued before the parameters/commands of Write Registers 1, 3, 5, (6 and 7 if used).

As a general rule, if changing from one operational configuration to another (e.g. 8-bit asynchronous to 7-bit asynchronous), the whole initialisation sequence should be repeated with the new parameters.

Write Register Summary

WR0	Write Register 0. Allows the programmer to perform a number of basic reset functions in addition to acting as the pointer to select the register for the next command/status transfer operation.
WR1	Write Register 1. Contains the bits which select the interrupt mode, and allows the interrupts to be enabled/masked.
WR2	Write Register 2. This register is programmed with the base address of an interrupt vector and is addressed through channel B only.
WR3	Write Register 3. The bits written to this register configure and control the channel receiver circuitry.
WR4	Write Register 4. This register is programmed with bits which select the mode of operation (asynchronous or synchronous), and some of the operational parameters of the selected mode.
WR5	Write Register 5. The bits written to this register configure and control the channel transmit circuitry.
WR6	Write Register 6. This register is programmed with a sync character in Monosync and Bisync, and a check address in the bit oriented synchronous modes.
WR7	Write Register 7. This register is programmed with a sync character in Monosync and Bisync, and a flag character in the bit oriented synchronous modes.

Write Register 0



Pointers (D0 to D2)

These bits signify the register for the next command/status transfer operation. If the next operation is a write, the pointer specifies a write register. If the next operation is a read, the pointer specifies a read register. Following a read or write to any register (except WR0), the pointer points to register 0 (i.e. Write Register 0 or Read Register 0).

Commands (D3 to D5)

These bits specify eight different commands, the function of which are detailed below.

1. Null Code. This command enables the programmer to specify the next register for a command/status transfer using the pointer bits, without affecting the operation of the SIO.
2. SDLC Send Abort. Used only in the bit oriented synchronous modes. The command causes the SIO to generate an abort sequence which informs the receiver of data from the SIO to terminate reception.
3. Reset Ext/Status Interrupts. After an External interrupt (caused by a change of state on a modem control input), or a Status interrupt (caused by detecting a break/abort condition in receive data or a transmit underrun/EOM condition in transmit data), corresponding status bits within Read Register 0 are latched. This command re-enables the bits and allows further interrupts to occur.
4. Channel Reset. This command performs the same function as a hardware reset but on the specified channel only. Issuing the command to channel A also resets the interrupt logic, clearing any current and all pending interrupts. All Write Registers in the channel must be reprogrammed, following the command.
5. Reset Rx Interrupt on First Character. If the Interrupt on First Receive Character Mode is in operation (bits 3 and 4 of WR1), an interrupt is generated on receiving the first character. At the end of the message, this command is issued to reactivate the mode.
6. Reset Tx Interrupt Pending. If the Transmit Interrupt is enabled (bit 1 of WR1), the SIO generates an interrupt every time the transmit buffer is empty. Issuing the Reset Tx Interrupt Pending command, resets the transmit interrupt, and prevents the interrupt being raised again until the transmit buffer is loaded with a character and becomes empty again.
7. Error Reset. Parity and Overrun errors are latched into Read Register 1 (the status register at the top of the receive error FIFO stack). These error conditions remain within the register until the Error Reset Command is issued.

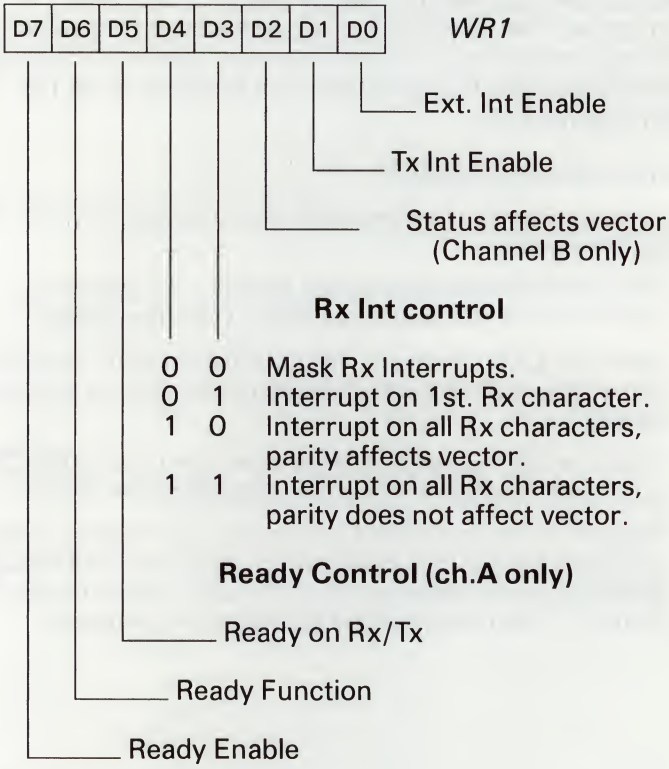
8. Return from Interrupt. This command is used to signify to the SIO that the interrupt routine in service specified by the last interrupt vector read by the CPU has been completed. The command is issued through channel A only and resets the in-service interrupt within the SIO, and allows the highest priority interrupt pending (if any) to interrupt the CPU.

Auxiliary Resets (D6, D7)

These bits specify four commands, the function of which are as described below.

1. Null Code performs the same function as previously described for the Null Code of the Command bits.
2. Reset Rx CRC Checker. This command resets the CRC error detection circuitry located in the channel receiver circuit.
3. Reset Tx CRC Generator. This command resets the CRC generator located in the channel transmitter circuit.
4. Reset Tx Underrun/EOM. When the SIO detects either a Transmit Underrun condition or the End of a Message (EOM), a corresponding bit is set within Read Register 0 (bit 6). The bit is reset by issuing this command.

Write Register 1



Ext. Int Enable (D0)

The logic state of this bit determines whether the External/Status interrupts are enabled (logic high) or masked (logic low). If enabled, an interrupt is produced as a result of the following conditions.

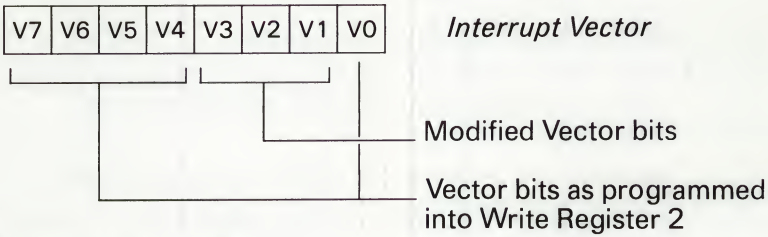
1. Transitions on the modem control lines; $\overline{\text{DCD}}$, $\overline{\text{CTS}}$, or $\overline{\text{SYNC}}$.
2. Detecting a break or abort condition in the receive data.
3. At the start of transmission of either CRC or sync characters, in synchronous modes, when the Transmit Underrun/EOM latch becomes set.

Tx Int Enable (D1)

The logic state on this bit determines whether the transmit interrupt is enabled (logic high) or masked (logic low). If enabled, an interrrupt is produced when the transmit buffer becomes empty.

Status Affects Vector (D2)

This bit is active only in channel B. If set to logic low, the interrupt vector programmed in Write Register 2 is written into the Read Register 2 in channel B unmodified. If set to logic high, the interrupt vector is modified according to the interrupting condition as detailed below.



Modified Vector Bits

V3	V2	V1	Interrupt Condition
0	0	0	Ch.B Transmit buffer empty
0	0	1	Ch.B Ext/status Interrupt
0	1	0	Ch.B Receive Character Available
0	1	1	Ch.B Special Receive Condition*
1	0	0	Ch.A Transmit buffer empty
1	0	1	Ch.A Ext/status Interrupt
1	1	0	Ch.A Receive Character Available
1	1	1	Ch.A Special Receive Condition*

* Special Receive Conditions include Parity error, Rx Overrun Error, Framing Error, End of Frame (SDLC).

Rx Int Control bits (D3, D4)

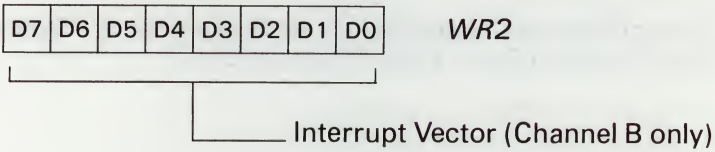
These two bits select the various receive conditions able to generate an interrupt.

1. Mask Rx Interrupts. Disables all receive interrupts.
2. Interrupt on 1st Rx Character. Enables an interrupt to be generated on detecting:
   (a) The first receive character in the receive buffer.
   (b) Any special receive condition.
3. Interrupt on all Rx Characters, parity affects vector. Enables an interrupt to be generated on detecting:
   (a) Any receive character within the receive buffer.
   (b) Any special receive condition.
4. Interrupt on all Rx Characters, parity does not affect vector. Enables an interrupt to be generated on detecting:
   (a) Any receive character within the receive buffer.
   (b) Any special receive condition apart from a Parity error.

Ready controls (D5 to D7)

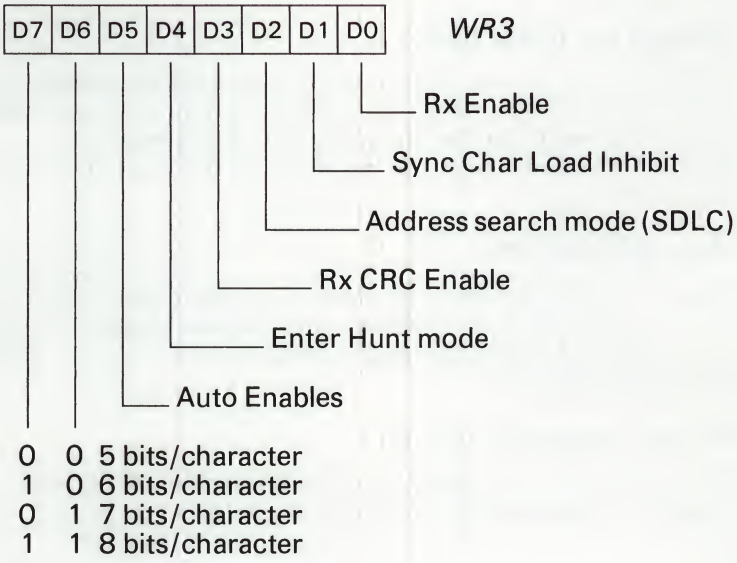
These three bits control the state of the $\overline{\text{W/RDYA}}$ output of channel A, which is not connected in the Portable. A W/RDYB output is not provided on a Z80 SIO/0, so these bits are also redundant in channel B. The state of the the three bits is thus immaterial.

Write Register 2



Write Register 2 is programmed with the base address for the interrupt vector read from Read Register 2 during the interrupt cycle. If the Status Affects Vector bit within Write Register 1 (WR1, D2) is set to logic high, the interrupt vector read by the programmer is the base address with the LSB modified as previously described. If the Status Affects Vector bit is set to logic low, the interrupt vector in the Read Register corresponds to the base address unmodified.

Write Register 3



Rx Enable (D0)

If set to logic high, the receiver circuits are enabled to accept data. If set to logic low, the receiver circuits are disabled.

Sync Character Load Inhibit (D1)

Setting this bit to logic high prevents any sync characters being loaded into the receive FIFO stack.

Address Search Mode (D2)

If a bit oriented synchronous mode is selected and this bit is set to logic high, only messages with an address (following the opening flag) matching either the programmed address in Write Register 6 or the global address (FFH), are accepted by the receiver circuits. If set to logic low, no address search is carried out.

Rx CRC Enable (D3)

This bit enables (logic high)/disables (logic low) the receiver CRC error detection circuits.

Enter Hunt Mode (D4)

The Hunt mode in synchronous modes is automatically entered following a reset. If character synchronisation is lost (any Sync mode) or the incoming data is not required (SDLC), the hunt phase can be re-entered by setting D4 to logic high.

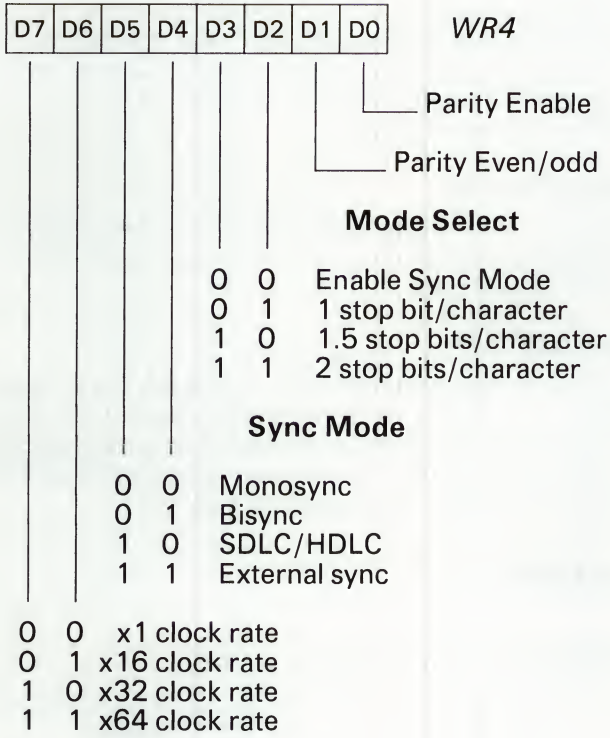
Auto Enables (D5)

Setting this bit to logic high allows the two modem input control lines, DCD and CTS to control data transfers over the serial data link. DCD controls the SIO receiver circuits, CTS the transmitter circuits.

Rx bits character (D6, D7)

The combination of these two bits determine the number of receive bits assembled to form a character.

Write Register 4



Parity Enable (D0)

If this bit is set to logic high, a parity bit is added to the transmit character data and is expected in receive data.

Parity Even/Odd (D1)

This bit is used when the Parity Enable bit is set to logic high to determine the sense of the parity code in the transmit data and the expected parity code in receive data. Even parity is signified by setting this bit to logic high, odd parity by setting the bit to logic low.

Mode Select (D2, D3)

The combination of these two bits differentiate between asynchronous and synchronous modes of operation and also specify the number of stop bits added to the transmit data in the asynchronous mode. The receiver circuits always check for one stop bit in the receive data, regardless of the number of stop bits in the transmit data.

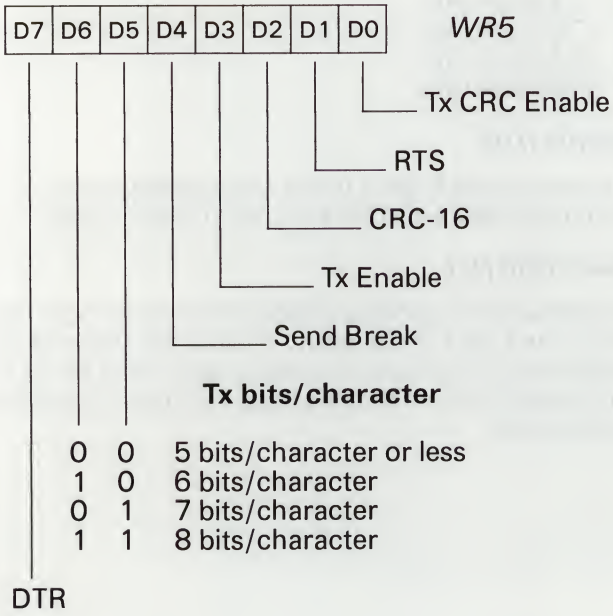
Sync Mode (D4, D5)

If synchronous mode is selected by the Mode Select bits, these bits determine the type of synchronous mode, for transmit and receive data.

Clock Rate (D6, D7)

These bits specify the multiplier applied to both the transmit and receiver input clock rate prior to being used to set the transmit and receive baud rates. For synchronous modes the x1 clock rate must be selected. Any rate may be selected for the asynchronous mode, apart from the x1 rate.

Write Register 5



Tx CRC Enable (D0)

This bit enables (logic high)/disables (logic low) the CRC generator in the transmit data path.

RTS (D1)

This bit controls the state of the modem control output Request to Send (RTS). When the RTS bit is set to logic high, the RTS output is set active (logic low/line spacing). When the RTS bit is reset to logic low, the RTS output is reset to the inactive logic high state (line marking).

In asynchronous mode, the RTS output is not reset by setting the RTS bit low, until after transmission of the character in the transmit buffer.

CRC-16 (D2)

This bit selects the CRC polynomial used by the transmitter and receiver circuits. If set to logic high, the CRC-16 polynomial ($X^{16} + X^{12} + X^5 + 1$) is selected. If set to logic low, the SDLC polynomial ($X^{16} + X^{15} + X^2 + 1$) is selected.

Tx Enable (D3)

This bit acts as the enable/disable control signal for the transmitter output. Character data in the transmit buffer cannot be transmitted or a break condition sent unless this bit is set to logic high. If set to logic low, the transmit output is held in the line idle marking condition.

Send Break (D4)

When set to logic high and the transmitter is enabled, the transmit data output is forced to the spacing condition, regardless of any data transmission in progress.

Tx bits/character (D5, D6)

The combination of these two bits specify the number of bits per character in the transmit data. Characters have to be assembled into the correct format by the programmer. All character lengths of less than 8 bits have to be right justified, with the following format.

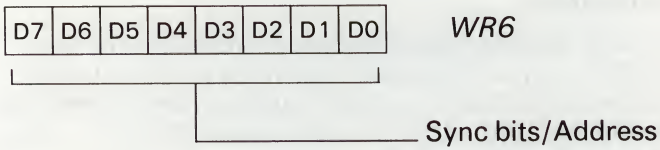
D7	D6	D5	D4	D3	D2	D1	D0	Bits/Character
0	D	D	D	D	D	D	D	7
0	0	D	D	D	D	D	D	6
0	0	0	D	D	D	D	D	5
1	0	0	0	D	D	D	D	4
1	1	0	0	0	D	D	D	3
1	1	1	0	0	0	D	D	2
1	1	1	1	0	0	0	D	1

D = Character data bits

DTR (D7)

This bit controls the modem control output Data Terminal Ready (DTR). When the DTR bit is set to logic high, the DTR output is set to the active state (logic low). When reset to logic low, the DTR output is reset to the inactive logic high state.

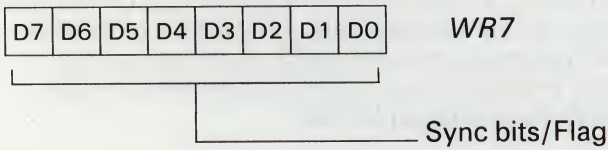
Write Register 6



Write Register 6 is programmed with:

(a) The transmit sync character in Monosync.
(b) The 8 LSB of the 16-bit sync character in Bisync.
(c) The 8-bit frame address byte in bit oriented modes.

Write Register 7



Write Register 7 is programmed with:

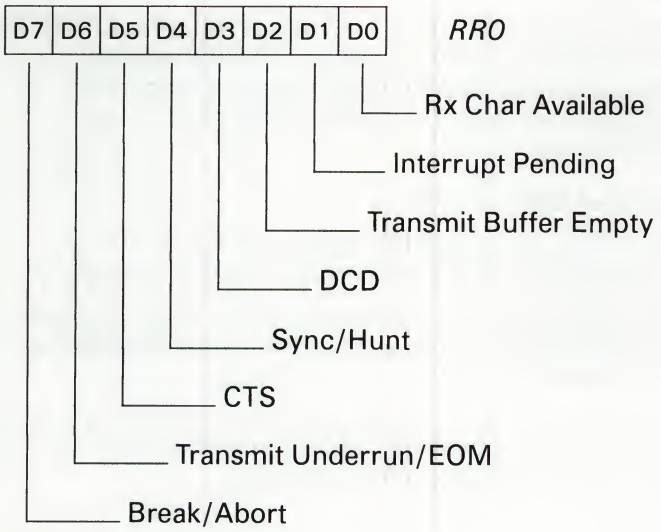
(a) The receive sync character in Monosync.
(b) The 8 MSB of the 16-bit sync character in Bisync.
(c) A flag character in bit oriented modes.

Read Register Definition

Both channels contain two Read Registers each (RR0 and RR1), which define the status within the channel. A third Read Register (RR2), accessed through channel B, contains a copy of the interrupt vector which indicates the SIO interrupt routine (if any) currently in service.

Reading the contents of Read Registers 1 and 2 requires a two byte transfer operation. The first, a write operation to the commands/status address location using the pointer bits of Write Register 0 to specify the Read Register: the second, the actual read operation from the same address to access the contents of the register. Following any read or write operation (apart from writing to Write Register 0), the address pointer within the SIO is always cleared to zero, allowing the contents of Read Register 0 to be accessed using a single read operation.

Read Register 0



Rx Char Available (D0)

This bit is held at logic high until all characters within the receive FIFO stack are read. Logic low indicates that no more receive characters are available.

Interrupt Pending (D1)

This bit only has significance in channel A. In channel B, the bit is always at logic low. When set to logic high, the bit indicates that an interrupt condition is present within the SIO.

Transmit Buffer Empty (D2)

This bit is set to logic high every time a character is transmitted out of the transmit buffer. It is reset to the logic low inactive state by reloading the transmit buffer with a new data character.

DCD (D3)

The DCD bit indicates the state of the modem control input Data Carrier Detect ($\overline{DCD}$) in channel B and the state of the modem control input $\overline{DSR}$ in Channel A. The state of the bit is latched every time any external/status interrupt condition occurs, and remains in the latched state until reset by writing the reset external/status interrupt command to Write Register 0.

Therefore, to ensure that the current state of the $\overline{DCD}/\overline{DSR}$ input is obtained, the bit should be read immediately following a reset external/status interrupt command. The $\overline{DCD}$ bit indicates the inverse of the state on the $\overline{DCD}/\overline{DSR}$ input.

Sync/Hunt (D4)

This bit only has any significance in channel B. The bit reflects the phase of the synchronous receive operation.

Initiating the Hunt phase causes the bit to be set to logic high. (i.e. On reset or setting the Enter Hunt Mode bit in Write Register 3).

On achieving character synchronisation, the bit is set to logic low as the receive phase begins and remains in this condition unless the Hunt phase is initialised again.

CTS (D5)

This bit functions in a similar manner with regard to the latching process, but indicates the inverse of the state on the Clear to Send input ($\overline{CTS}$).

Transmit Underrun/EOM (D6)

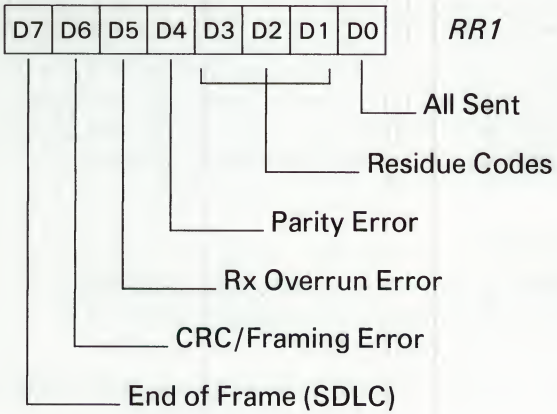
In synchronous modes, the bit is set to logic high following a system/channel reset, allowing sync/flag characters to be sent when the transmit buffer becomes empty. When the reset transmit underrun/EOM command is issued to Write Register 0, the transmit underrun/EOM bit is set to logic low. This enables CRC characters to be automatically sent instead of the sync/flag characters.

Break/Abort (D7)

In asynchronous modes, this bit is set to logic high, when a break is detected in the receive data. The bit is not reset until, a reset external/status command is issued to Write Register 0 and the break condition is removed.

In the bit oriented modes, the bit is set to logic high on detecting an abort sequence in the receive data. The bit is reset to logic low on loading Write Register 0 with the reset external/status command. The bit is not used in the byte oriented synchronous modes.

Read Register 1



All Sent (D0)

In asynchronous modes, this bit is set to logic high when all the bits of the character have been transmitted onto the serial link.

In synchronous modes the bit is permanently set to logic high.

Residue Codes (D1 to D3)

The combination of these three bits indicate the length of the I-field in the bit oriented modes where the I-field is not an integral multiple of the character length.

Parity Error (D4)

When parity is enabled, this bit is set to logic high on detecting a receive character whose parity does not match the sense programmed by bit 1 of Write Register 4. The bit remains set in the error condition until reset by loading the error reset command into Write Register 0.

Rx Overrun Error (D5)

This bit is set to logic high when one or more receive characters have been overwritten in the receive FIFO buffer. The bit remains set in the error condition until reset by loading the error reset command into Write Register 0.

CRC/Framing Error (D6)

The function of this bit is dependent on the mode selected. In asynchronous modes, the bit is set to logic high on detecting a receive character with incorrect stop bits (framing error). The error condition only persists for the particular character stored in the receive buffer.

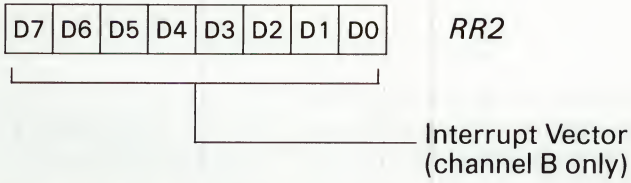
In synchronous modes, the bit indicates the result from the receive CRC error detection circuit. A logic high indicates a CRC error. The error conditions are reset to the inactive low state after issuing the error reset command to Write Register 0.

End of Frame (D7)

In the bit oriented modes, this bit indicates that a valid closing flag has been detected and the CRC Error and residue codes are now valid.

The bit is reset by issuing an error reset command to WR0.

Read Register 2



Read Register 2 contains the interrupt vector and is read through channel B only. If the status affect vector bit is set (D2, Write Register 1), the register indicates the current interrupt service routine (if any) in operation.

If no interrupts are pending, the vector is set to the condition for a special receive condition in channel B (see Write Register 1 description). If the status affect vector bit is not set (logic low), the register contains a copy of the vector written into Write Register 2 in channel B.

SIO Interrupt Sequence

The SIO incorporates an elaborate interrupt structure, which acts as an extension to the interrupt structure provided by the System Interrupt Controller (PIC). A single interrupt output line (INT) connects the SIO to the PIC.

Both channels within the SIO are able to generate an interrupt for a variety of conditions. All interrupting sources can be enabled or disabled (masked) by software, and are ordered on a priority basis.

All sources within Channel A are assigned a higher priority than Channel B. Within each channel, the assigned priority is as detailed below:

1. Special receive condition (Highest).
2. Receive character available.
3. External/status interrupt.
4. Transmit data required (Lowest).

If a SIO interrupt is in service when a higher priority interrupt condition occurs, the higher priority condition is granted service. The SIO stores the lower priority condition and resumes the interrupt service routine on completion of the higher priority interrupt routine.

A special receive interrupt is generated when the SIO detects any of the following conditions in the receive data:

1. Parity errors.
2. Receiver overrun.
3. Framing errors (asynchronous modes).
4. End of Frame (SDLC/HDLC only).

Receive character available interrupts can be programmed to generate an interrupt on every receive character or for the first character only (synchronous modes).

An external/status interrupt can be caused by any of the following conditions:

1. Transitions on the modem control lines; $\overline{DCD}$, $\overline{CTS}$ and $\overline{SYNC}$.
2. A break in receive data (asynchronous modes).
3. An abort sequence in receive data (bit synchronous modes).
4. Transmit underrun/EOM condition in transmit data (synchronous modes).

The actual sequence of events performed on the occurrence of a SIO interrupt condition is as follows.

The SIO compares the new interrupt with any interrupt currently in service. If the new interrupt is of a higher priority than the current interrupt, the $\overline{INT}$ output is set active (logic low). If the new interrupt is of a lower priority, the interrupt condition is stored until it becomes the highest priority.

The $\overline{INT}$ output, after inversion, forms the interrupt request line (IR4) to the PIC.

Providing IR4 is the highest priority unmasked interrupt request to the PIC, an interrupt is issued to the CPU.

The CPU determines the device generating the interrupt by performing a PIC interrupt acknowledge sequence, which then provides an associated interrupt vector, to specify the SIO as the source of interrupt (vector 54H).

The programmer is then able to determine the actual SIO process generating the interrupt by reading the interrupt vector stored in the SIO.

This interrupt vector is constructed from a base address previously supplied to the SIO modified according to the condition generating the interrupt.

The SIO interrupt thus allows the programmer to select the appropriate interrupt service routine.

On completion of the SIO service routine, the CPU has to issue a return from interrupt command (to WR0), to enable any lesser priority pending interrupts to generate an active interrupt output.

The SIO is able to produce eight different interrupt vectors according to the type of interrupt as detailed on the previous pages (4 vectors for each channel). In the case where the interrupt can be caused by a number of different conditions (i.e. special receive and external/status), the actual cause can be detected by reading bits within the appropriate status register.

Keyboard/Mouse Data

General

Transfer of data from the keyboard/mouse is handled by channel B of the SIO. The channel is programmed to support uni-directional synchronous communications at a fixed data rate (approximately 3.85 kbits per second).

The format of the data transmitted from both the keyboard and mouse is identical, both consisting of a 4 byte synchronous packet. The first byte in the packet is the sync byte (5AH). The remaining three contain data. (This is true in all cases apart from when the Keyboard System Reset button is pressed. In this case, contiguous sync bytes are transmitted to the Systems Unit instead. The function of this is to generate a hardware reset. This mechanism is discussed in the System Detail chapter).

The clock for the synchronous data is inherent in the data stream transmitted to the Systems Unit. This is split into separate Monosync data and clock waveforms by the IR Receiver Circuiting on the Interface board.

Reception of the incoming synchronous data consists of two separate phases. The first phase is the Hunt phase, where channel B of the SIO analyses all incoming data, searching for the sync byte 5AH. This is the header byte for all valid keyboard and mouse transmissions.

Detecting the sync byte automatically switches the SIO into the second phase, the Receive phase, where the three bytes of data following the header are loaded into the three-byte buffer of Channel B.

Every time the SIO tranfers received data into the top register of the three-byte buffer, it generates an interrupt to the CPU (via the PIC).

The Systems Unit does not exercise control of data flow over the IR link (all transfer of information is one-way only). To prevent any loss of data, the CPU has to read the incoming data at a fast enough rate before it is overwritten in the receive FIFO stack.

The software must therefore be capable of processing data stored in the Channel B SIO receive data buffer at the maximum possible incoming receive data rate. This corresponds to the minimum time taken between the end of one packet and the beginning of the next consecutive four byte packet from the Keyboard and is of the order of 20 ms.

Encoding the data into a synchronous packet format provides a high degree of protection to interference from stray infra-red transmissions generated by other sources. It prevents the system being unnecessarily interrupted by other sources, since only transmissions which contain a sync byte of 5AH will be considered to be valid.

Further protection to sources of interference is provided in the coding of the bytes following the sync header. These three bytes are encoded using a Hamming format. This is totally transparent to the SIO. The SIO is only concerned with receiving bytes of data and is not concerned with the make-up of the bytes. All "de-hamming" is carried out by the BIOS.

A description of the format, character codes and their significance, supplied from the Keyboard is detailed in the Keyboard chapter. Details of the Mouse and its transmission format is packaged into a separate manual.

Channel B Programming Details

In addition to providing a series of Write Registers to configure the SIO to interface with the keyboard and mouse, one other parameter is programmed through channel B. This is the base address for the SIO interrupt vector.

Copy Registers

As the write registers are write-only and cannot be read, the BIOS always keeps a copy of certain SIO registers which may be of use to programmers. These are the registers WR1, WR3, WR4 and WR5 in channel B.

The register copies are located in RAM in a register copy table along with copies of other write only registers used in the system. The reason for keeping copies is to avoid the possibility of adverse effects arising when the BIOS and the application programmer both update bits within the register. Since either source could unknowingly modify other bits within a particular register which are vital for the other's operation, a programming structure has been devised to minimise this problem.

The BIOS updates the write register by performing the following sequence:

1. Reads in the data byte stored in the copy register.
2. Modifies the bits within the byte as appropriate, leaving all other bits untouched.
3. Writes the modified byte to the Write Register.
4. Updates the copy register.

To avoid unintentionally overwriting bits set by BIOS, the same approach should also be adopted by the application programmer when directly accessing the hardware.

A copy of a particular register is located by:

1. Reading a double word pointer located at absolute address location 722H (the start of the register copy table).
2. Adding an offset to the pointer to form the address of the register copy byte of interest.

The offset for the Channel B copy registers are as follows:

Write Register 1	04H
Write Register 3	05H
Write Register 4	06H
Write Register 5	07H

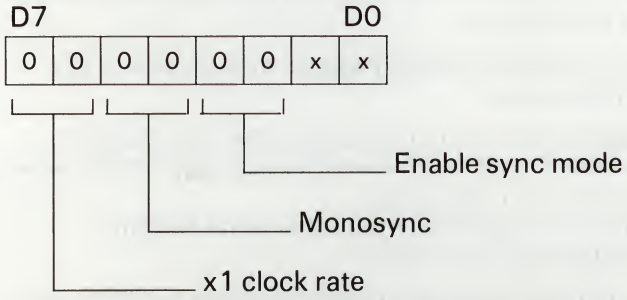
Initialisation

The parameters issued to the Write Registers of channel B for interfacing the SIO to the Keyboard, and the interrupt vector base address are programmed during an initialisation routine by the BIOS.

Once channel B is initialised with the keyboard parameters and the base address for the interrupt vector (00H to Write Register 2), transfer of data between the Keyboard and the processors can proceed via the synchronous hunt/receive mechanism.

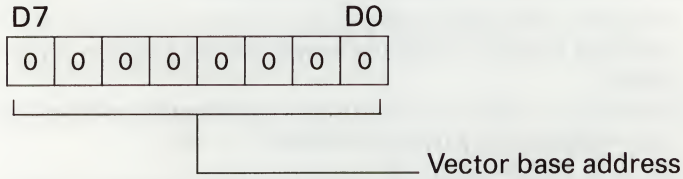
The format of the command bytes supplied to the Write Registers of channel B during initialisation are detailed in the following pages. The address locations utilised by channel B are described in a previous section, under the heading "Processor Interface".

Write Register 4

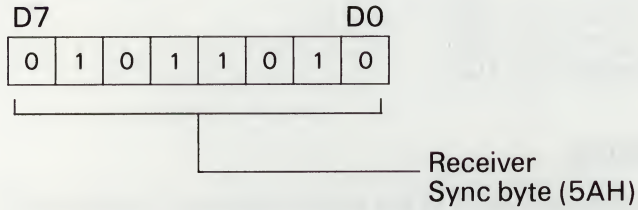


x = not applicable to mode and application, program to 0.

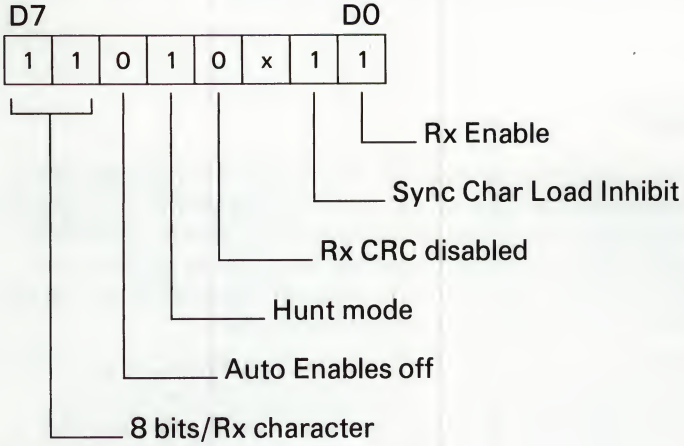
Write Register 2



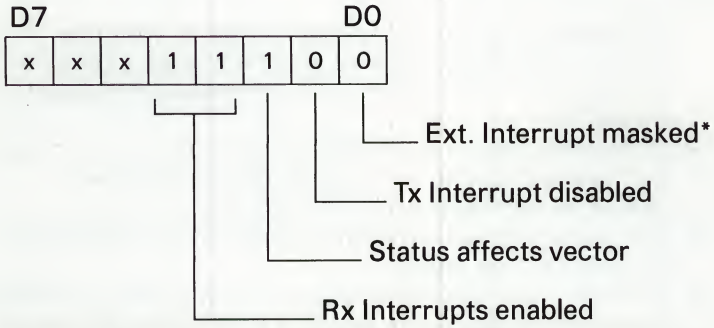
Write Register 7



Write Register 3



Write Register 1



* unmasked to allow the RS232C line $\overline{\text{DCD}}$ to generate an interrupt

RS232C Communications

General

The interface to the RS232C serial link is implemented by channel A of the SIO, which can be programmed to operate in either asynchronous or synchronous communications modes with transmit and receive baud rates determined either via the Programmable Interval Timer (PIT), or via the external data communications equipment.

The clock inputs to the SIO pins TxCA and RxCA, are used to determine the transmit and receive baud rates for the RS232C link and are supplied via the data selector. The select lines to the data selector are provided by two bit-wide control ports mapped in the system I/O space. The ports allow the programmer to select:

1. Different rates for transmit and receive, with the receive rate determined by the clock output OUT1 of the Interval Timer and the transmit rate determined by OUT2.
2. Different rates for transmit and receive as determined by the external data communications equipment (RS232C connector pins, Rc and Tc).
3. The same rate for transmit and receive as determined by OUT2 of the timer.
4. Different rates for transmit and receive with the receive rate determined by the external data communications equipment (RS232C pin Rc) and the transmit rate determined by OUT2 of the timer.

Details of the timer and the appropriate programming values to produce the majority of the commonly used baud rates are detailed in the TIMER chapter. These values apply to asynchronous communications only and require channel A to be programmed in the x16 clock rate mode to achieve the correct baud rate.

One of the values detailed in the TIMER chapter produces a baud rate value which is marginally outside the +/— 5% tolerance normally allowed for communications over an RS232C link. This is the highest baud rate setting, 19200 bauds which should therefore be used only for Apricot-to-Apricot communications.

The selected transmit rate clock (TxCA) is also supplied to the RS232C connector (connector pin, TxClk), for use by the external data communications equipment.

Four of the most commonly used RS232C modem control lines are connected to various inputs and outputs of channel A of the SIO and are available for coordinating data transfers over the serial link.

These include the outputs $\overline{\text{RTS}}$ and $\overline{\text{DTR}}$, and the inputs $\overline{\text{DSR}}$ and $\overline{\text{CTS}}$. The modem control input $\overline{\text{DCD}}$ is accessible through channel B. Use of each individual modem control line is dependent on compatible facilities within the external equipment.

Generation of the output modem control lines and status monitoring of the input modem control lines is directly under program control. The SIO can be programmed to generate an interrupt to the CPU every time a transition occurs on any of the input control lines.

RS232C Connector Detail

The definition of the available RS232C connections provided by the 25-way D-type connector are detailed in the following table.

The RS232C voltage levels are defined below. The line idle condition is indicated by continuous marking.

1. Line marking; Between —3V and —15V relative to 0V.
2. Line spacing; Between +3V and +15V relative to 0V.

RS232C Connector Pin Definition

Pin	Description	Pin	Description
1	Frame Ground	9 to 14	N.C.
2	Tx Data	15	Tc (input)
3	Rx Data	16, 18 , 19	N.C
4	RTS	17	Rc (input)
5	CTS	20	DTR
6	DSR	21 to 23	N.C.
7	0V (Signal GND)	24	TxClk (output)
8	DCD	25	N.C.

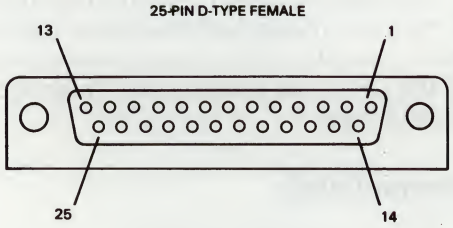


Figure 2. RS232C Connector Detail.

Channel A Programming Details

The following paragraphs detail the format of the Write Registers for setting channel A to operate in the asynchronous mode and also the various options available for controlling the transfer of data over the serial link. For details of programming the channel for the various synchronous communication modes, reference should be made to the appropriate Zilog documentation.

The address locations utilised by channel A and the method for writing data to the Write Registers are described in a previous section under the heading "Processor Interface".

Copy Registers

As with channel B, the BIOS also keeps a copy of certain SIO registers within channel A which may be of use to programmers. These are the registers WR1, WR3, WR4 and WR5.

The reason for keeping copies is to prevent contention arising between the BIOS and an application programmer when updating write-only registers. The mechanism for writing to the SIO registers (as described in the Keyboard copy registers) should also be adopted for channel A.

The offset from the double-word pointer for the channel A copy registers are as follows:

Write Register 1	00H
Write Register 3	01H
Write Register 4	02H
Write Register 5	03H

Asynchronous Communications

To receive and transmit data in the asynchronous mode, the following conditions have to be determined:

1. The transmit and receive baud rates.
2. The character length.
3. The number of stop bits in the transmit character (1 stop bit is always expected in the receive data).
4. The sense of the parity, if any.
5. The interrupt mode.
6. The line protocol.

Note: The modem control line DCD from the RS232C interface is connected to DCDB of channel B.

Baud Rate Selection

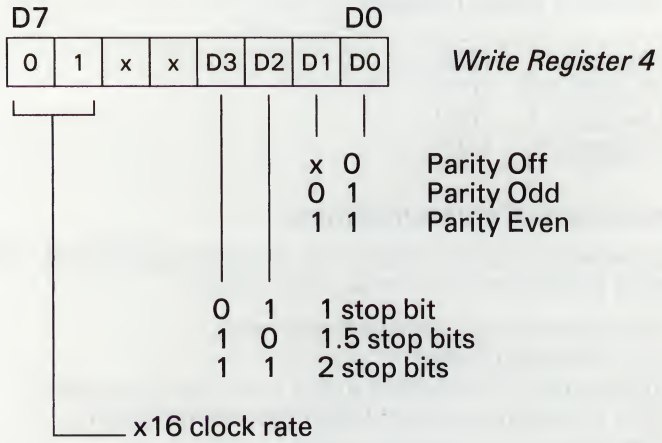
The clock source for determining the transmit and receive baud rates is selected according to the programmed state of two bit-wide control ports mapped at I/O address locations 3CH (select line L6) and 3EH (select line L7).

Both ports are write-only and are wired to data bus line D0 and therefore only use bit 0. The logic state of all the other bits written to the port are thus immaterial. The status of the two select lines directly reflect the status of the bits written to the ports. This is as detailed below.

L7	L6	Rx Clock	Tx Clock
0	0	OUT1 (TMR)	OUT2 (TMR)
0	1	Ext. Clock (Rc)	Ext. Clock (Tc)
1	0	OUT2 (TMR)	OUT2 (TMR)
1	1	Ext. Clock (Rc)	OUT 2 (TMR)

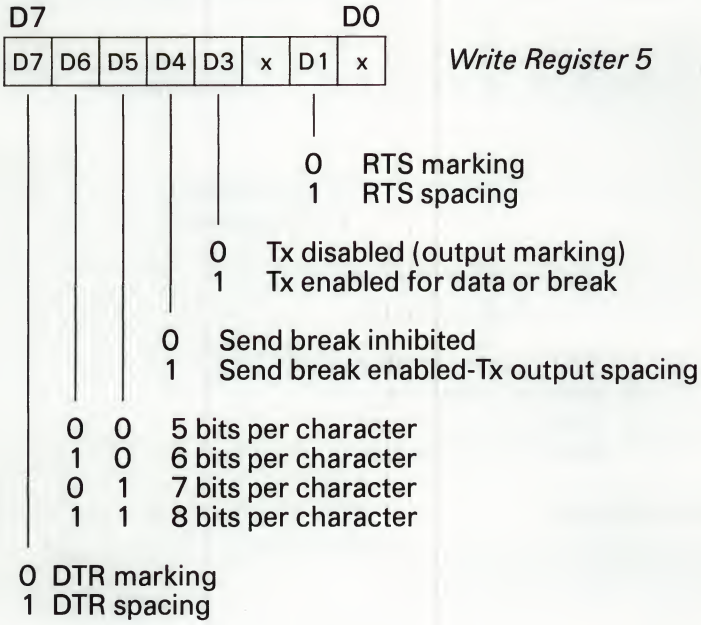
Write Register 4

This register determines; the number of stop bits in the transmit data, the sense of the parity (if any) in the transmit data and the expected parity in the receive data, and the multiplier applied to both the transmit and receive input clock rates prior to setting the baud rates.



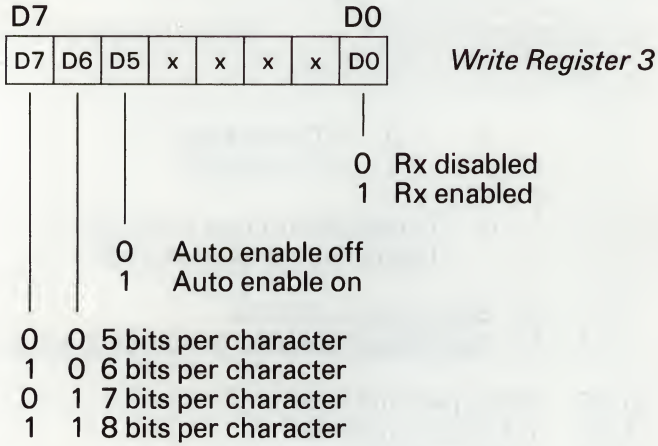
Write Register 5

This register determines the transmit character length, controls the modem outputs RTS and DTR, allows the programmer to enable/disable transmission, and also generate a break in transmission.



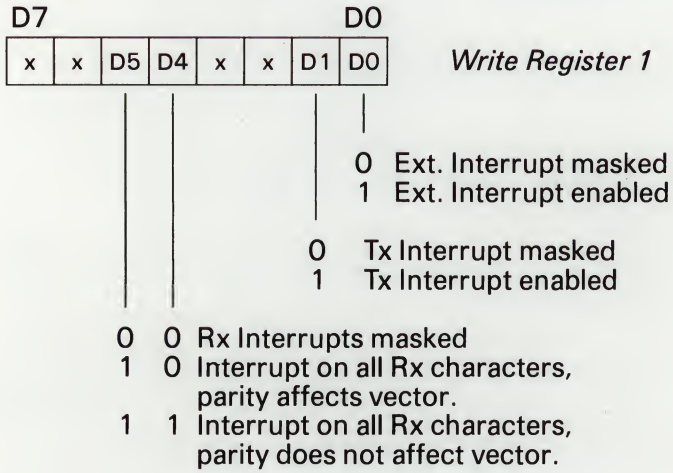
Write Register 3

This register determines the number of receive bits assembled to form a character, allows the programmer to enable/disable reception and also selects whether the modem control lines CTS and DSR control data transmission and reception over the serial link (Auto Enable).



Write Register 1

This register enables/disables the transmit, receive and external/status interrupts.



SIO Pin Detail

The two SIO communication channels are designated channel A (used for the RS232C link) and channel B (used for the keyboard link). On the block diagram at the beginning of the chapter, connections to channel A are denoted by the suffix A (e.g. TxDA), and to channel B by the suffix B (e.g. RxDB). All the remaining inputs and outputs of the SIO are connections to the processor. A description of each pin is detailed on the following pages.

System Connections

D0 to D7	Data bus, used to transfer data and commands between the processor and the SIO.
$\overline{\text{IORQ}}$	Input/Output Request. Control input, active state logic low, derived by combining the read and write commands from the system control bus ($\overline{\text{IOWR}}$ and $\overline{\text{IORD}}$) using an AND gate. Used in conjunction with $\overline{\text{RD}}$ and the address inputs B/$\overline{\text{A}}$, C/$\overline{\text{D}}$ and $\overline{\text{CE}}$ to control the transfer of data and commands between the SIO and the processors. An active state on $\overline{\text{IORQ}}$ indicates a transfer operation on the data bus; $\overline{\text{RD}}$ signifies the direction of data transfer (see below).
$\overline{\text{MI}}$	Machine Cycle 1. Control input, active state logic low. Function not used.
$\overline{\text{RD}}$	Read. Control input. Used in conjunction with $\overline{\text{IORQ}}$ and the address inputs B/$\overline{\text{A}}$, C/$\overline{\text{D}}$ and $\overline{\text{CE}}$ to control the transfer of data and commands between the SIO and processing elements. If both $\overline{\text{IORQ}}$ and $\overline{\text{RD}}$ are set low and a valid address is set up, the direction of data/command transfer is from the SIO (read operation). If $\overline{\text{IORQ}}$ is set low and $\overline{\text{RD}}$ is set high, with a valid address set up, the direction of data/command transfer is to the SIO (write operation).
$\overline{\text{CE}}$	Chip Enable. Address input, active state logic low. When active, indicates that the SIO is selected for a data/command transfer operation.

B/$\overline{A}$	Channel B/Channel A select. Address input. A logic low on B/$\overline{A}$ with $\overline{CE}$ also set low indicates that the data/command transfer operation involves channel A. A logic high on B/$\overline{A}$ with $\overline{CE}$ set low indicates that the data/command transfer operation involves channel B.
C/$\overline{D}$	Control/$\overline{Data}$ select. Address input. A logic low on C/$\overline{D}$ with $\overline{CE}$ also set low indicates that the information on the data bus is interpreted as data. A logic high on C/$\overline{D}$ with $\overline{CE}$ set low indicates that the information on the data bus is interpreted as commands.
CLK	System clock input. 2.5 MHz clock with a 50% duty cycle for internal timing within the SIO.
$\overline{RESET}$	System reset. Input, active low. When active, disables the receive and transmit sections of both channels and sets the transmit lines to the marking condition, requiring the registers within the SIO to be re-initialized before performing data transfer operations.
$\overline{INT}$	Interrupt request. Output, active low. Set active when an interrupt condition is detected internally within the SIO.

Channel A Connections

TxDA	Transmit Data. Serial data output to the RS232C interface via a line driver. A mark is indicated by logic high on TxDA and a space, by logic low.
RxDA	Receive Data. Serial data input from the RS232C interface via a line receiver. A mark is indicated by a logic high and a space, by logic low.
TxCA	Transmitter Clock. Input used to determine the transmit baud rate of the RS232C channel. When operating in asynchronous mode, a facility exists within the SIO to divide the input clock frequency internally, prior to being used to set the baud rate. The divider is controlled by software and allows serial data to be transmitted at a rate of 1, 1/16th, 1/32nd or 1/64th of the rate supplied to TxCA. In synchronous modes, this facility is not available, so that TxCA corresponds directly to the transmit baud rate.

RxCA	Receiver Clock. Input used to set the receive baud rate of the internal data receiver circuitry. In asynchronous modes, the receiver clock is divided by the same divisor programmed for the transmitter clock. In synchronous modes, RxCA directly sets the receive baud rate of the data receiver circuitry.
$\overline{\text{DCDA}}$	Data Carrier Detect A. Control input from the RS232C interface via a line receiver, connected to the Data Set Ready line - $\overline{\text{DSR}}$. When used with a modem, the active state on $\overline{\text{DSR}}$ indicates that the modem has data to send. The input can be programmed to operate in one of two modes. The first mode allows $\overline{\text{DSR}}$ to act as the enable control line to the receive section of channel A. The second mode sets a control bit within an internal register according to the state of $\overline{\text{DSR}}$ and also has the facility to generate an interrupt request to the Interrupt Controller, every $\overline{\text{DSR}}$ transition.
$\overline{\text{DTRA}}$	Data Terminal Ready. DTR control output to the RS232C interface via a line driver. When used with a modem, the active state (logic low) indicates that the SIO is ready to start handling data. $\overline{\text{DTRA}}$ is controlled by software, following the logic state of an associated control bit within an internal register.
$\overline{\text{RTSA}}$	Request to send. RTS control output to the RS232C interface via a line driver. When used with a modem, the active state (logic low) indicates that the SIO has data ready to send. $\overline{\text{RTSA}}$ is controlled by software, following the logic state of an associated control bit within an internal register.
$\overline{\text{CTSA}}$	Clear to send. CTS control input from the RS232C interface via a line receiver. When used with a modem, the active state (logic low) indicates that the modem is ready to receive data. $\overline{\text{CTSA}}$ can be programmed to operate in one of two modes. The first mode allows $\overline{\text{CTSA}}$ to act as the enable control line to the transmit section of channel A. The second mode sets an associated control bit within an internal register according to the state of $\overline{\text{CTSA}}$ and also has the facility to generate an interrupt request to the Interrupt Controller, every time a transition occurs on $\overline{\text{CTSA}}$.

Channel B Connections

RxDB	Receive Data. Serial data input for data from the Keyboard and mouse. Decoded infra-red signals supplied via the IR Receiver Board.
RxTxCB	Receiver Clock for channel B. Sets the receive rate of the internal receiver circuitry. Derived from the incoming IR data stream supplied from the IR Receiver Board.
$\overline{\text{SYNCB}}$	Synchronisation. Control output used to generate a hardware system reset. Everytime the SIO receives a sync character, the SIO pulses this pin low. When the Keyboard reset key is pressed, the keyboard transmits a contiguous stream of sync characters which results in pulses being produced in a regular 50% duty cycle. This causes a capacitor to gradually charge. If the key is held down for approximately one second, the capacitor is charged sufficiently to change the state of a trigger circuit which causes a system reset. This mechanism is discussed more fully in the System Details chapter.
$\overline{\text{DCDB}}$	Data Carrier Detect B. Control input from the RS232C interface via a line receiver, connected to the Data Carrier Detect line - DCD. When used with a modem, the active state on DCD indicates that the modem has detected data sent to it. The input sets a control bit within an internal register according to the state of $\overline{\text{DCDB}}$ and also has the facility to generate an interrupt request to the Interrupt Controller, on every $\overline{\text{DCD}}$ transition.

Parallel Printer Interface 2.8

Contents

Illustrations

Introduction

The Parallel Printer Port forms the interface for parallel printers and plotters and is located on the Interface Board. The circuitry consists of a number of 74LS series logic elements and a Centronics connector.

The Connector is accessible to the user when the Expansion cover on the back of the Systems Unit is removed. Cut-outs are provided in the cover panel, which when in position, neatly and tidily routes the printer cable out of the back of the machine.

The pin connections to the Centronics port support the standard Centronics 8-bit data interface and five of the more common handshake signals required/supplied by parallel printers.

These include:

1. Data Strobe
2. Busy
3. Printer Select
4. Fault
5. Paper Empty

The interface also provides regulated dc supply outputs for use by the ACT thermal printer.

Details

General

The printer interface is constructed from; an octal D-Type latch, an octal buffer, an address decoder, a Centronics connector and a few other assorted LS components. A block diagram of the circuitry is shown on Figure 1.

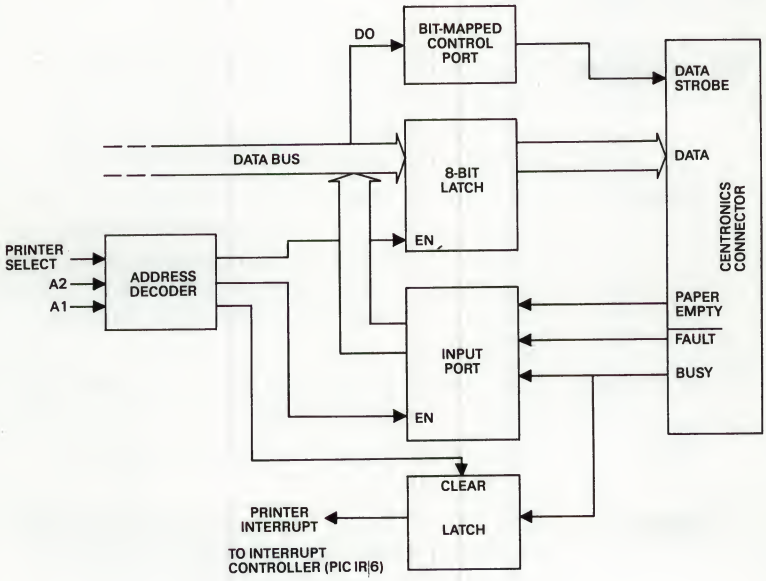


Figure 1. Parallel Printer Interface

Data Interface

The octal D-type latch forms the interface between the system data bus and the 8-bit data output to the printer. It is located at 20H in the system I/O space. The programmer updates the data sent to the printer by simply sending bytes to this address.

Control and Status

Five control lines are connected to the Centronics output which are used to control the transfer of data between the computer and the printer, and to provide the programmer with information on the printer status. The function of each line is as detailed below.

Data Strobe	Output timing signal. Used to latch the output data into the printer. Normally at logic high. Positive edge of the logic low pulse is used to latch data into the printer.
Busy	Input signal. Logic high state indicates that the printer is unable to receive any data; logic low indicates that the printer is able to receive data. The input generates an interrupt on changing state from Busy to Not Busy.
Paper Empty	Input signal. Logic high state indicates that the printer is out of paper.
$\overline{\text{Fault}}$	Input signal. Logic low state indicates a printer fault condition.
Select	Input signal. Logic high state indicates that the printer select status is active (selected and ready for operation); logic low that the select status is inactive (printer not selected and thus not ready for a data transfer).

The data strobe output is generated by the programmer writing data to the I/O port located at address 3AH. This is a bit wide port formed by a bit addressable latch which uses the LSB data line D0 only.

Writing 00H to the port sets the data strobe output high. Writing FFH to the port sets the data strobe output low. A positive edge on data strobe is required to latch data into the printer. This is produced by toggling the state of the data strobe output by writing FFH followed by 00H to I/O address 3AH.

The Busy input is wired to the system data bus line D0 via an octal buffer and also to a latch which is connected to the interrupt request line IR6 of the interrupt controller (the PIC located on the Processor Board). The interrupt line is asserted by a change on the input line from Busy status to Not Busy (logic high to logic low transition).

The programmer is able to use this interrupt to determine that the printer is ready to receive another byte of data. The normal operation of a printer is to assert Busy immediately a data byte is received to allow the printer time to print the character sent, or action the command (depending on the codes sent).

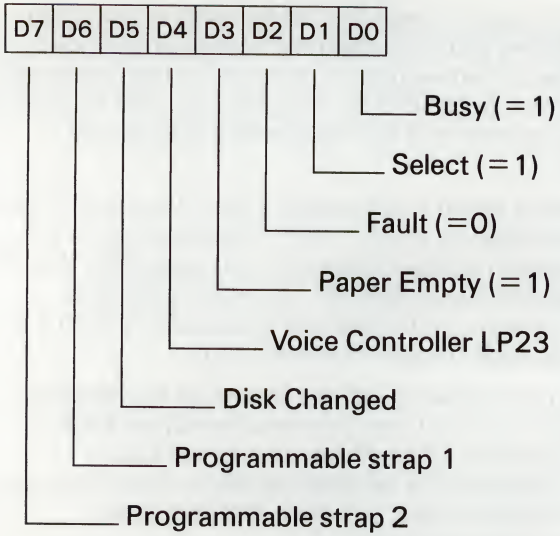
The printer resets the Busy input to the inactive state following completion of the printer operation to signify that another byte of data can be sent. This change of state from Busy to Not Busy generates the interrupt.

The interrupt line is not automatically reset by responding to the interrupt. The programmer has to reset the line by writing to address location 22H in the system I/O space. No specific data is required; the interrupt line is reset to the inactive state by the action of simply writing to the address.

The programmer can check the status of the printer Busy line by reading bit 0 of the octal buffer. This is a byte wide read only port mapped in the system I/O space at address location 24H. The data on D0 from this port mirrors the state of the Busy line.

The input lines Select, Fault and Paper Empty provide an indication of the printer status and are also connected to the octal buffer. The four printer status inputs can be checked by reading the four least significant bits from I/O address location 24H.

Printer status line to status bit mapping is as detailed below with the state on the four bits corresponding directly to the logic state on the input line.



(The four most significant bits of the data byte are illustrated for clarity. These are not used to signify any state associated with the printer. They are used to monitor various conditions on the board which are described in other chapters).

Register Summary

I/O Address	Function
20H	Printer Data
22H	Reset Printer Busy Interrupt
24H	Printer status
3AH	Data strobe

Centronics Interface Details

The definition of the control outputs to and from the printer and details of the Centronics connector are provided below.

Pin	Description	Pin	Description
1	Data strobe	19	0V
2	PD0 — 8-bit data output	20	0V
3	PD1	21	0V
4	PD2	22	0V
5	PD3	23	0V
6	PD4	24	0V
7	PD5	25	0V
8	PD6	26	0V
9	PD7	27	0V
10	N.C.	28	0V
11	Busy	29	0V
12	Paper Empty (PE)	30	0V
13	Select	31	N.C.
14	N.C.	32	Fault
15	N.C.	33	0V
16	0V	34	+12V @ 20 mA max.
17	Frame Ground	35	+5V @ 100 mA max.
18	N.C.	36	N.C.

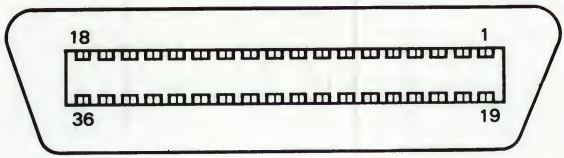


Figure 2. Centronics Connector pin detail

Timer 2.9

Contents

Illustrations

Introduction

The Intel 8253-5 Programmable Interval Timer (PIT) is located on the Interface Board. The timer utilizes a 2 Mhz clock output from the 6301 processor to generate:

1. A clock output (OUT0), which is connected to an interrupt request line (IR0) of the Interrupt Controller (PIC) on the CPU Board. The output is used by the BIOS as a general purpose system timer (20 ms).
2. Two squarewave clock outputs (OUT1 and OUT2) which can be used to set the baud rates for the RS232C serial interface. The frequencies of the two clock outputs are under direct control of the programmer.

Note: The squarewave output from the Timer (OUT1) is used by both the RS232C interface (for implementing a split rate baud clock) and the speech interface circuitry. If split rate baud rate clocks are required, the speech interface cannot be successfully utilised. This also applies to the opposite situation.

Details

General

A block diagram showing how the Timer circuit fits into the system is illustrated below.

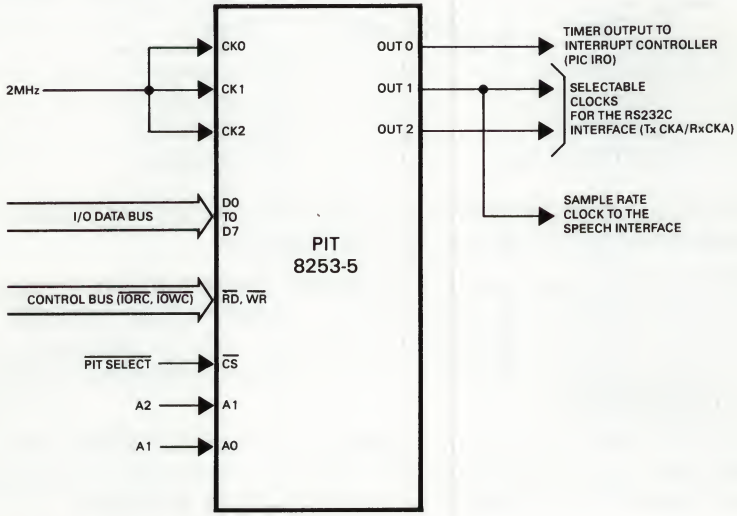


Figure 1. Timer block diagram

TMR Pin Definition

CK0	Clock input for Counter 0
CK1	Clock input for Counter 1
CK2	Clock input for Counter 2
OUT 0	Output from Counter 0
OUT 1	Output from Counter 1
OUT 2	Output from Counter 2
D0 to D7	Data bus connection
$\overline{RD}$	Read control line
$\overline{WR}$	Write control line
$\overline{CS}$	Chip select input
A0,A1	System address bus inputs

The timer is organized internally as three independent programmable 16-bit counters. Each counter can be set into various operating modes by programming data into a control register. On set up, counters count down on the negative edge of the 2 MHz clock input.

The system software views the three counters and the control register as an array of input/output ports. The port address locations defined by the timer select and system address bus connections are detailed below:

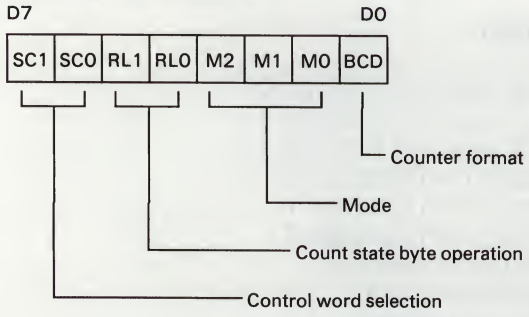
Port	Address	Data
Counter 0	08H	Count state 0
Counter 1	0AH	Count state 1
Counter 2	0CH	Count state 2
Control register	0EH	Control byte

Data can be written to all four address locations but can be read only from the three counter locations.

Each counter has to be initialized with the required mode of operation, the count state format, and the number of bytes in the count state using the control byte, prior to loading the initial count state. Bits in the control byte specify the counter to be addressed.

The counters begin counting downwards on completion of the count state load operation. The count state can be one or two bytes. The format of the control word is as follows:

Control Word Format



Control word selection

SC1	SC2	
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Invalid

Count state byte operation

RL1	RL2	
0	0	Latch count state into storage register
0	1	Read/load most significant byte only
1	0	Read/load least significant byte only
1	1	Read/load least significant byte followed by most significant byte.

Mode

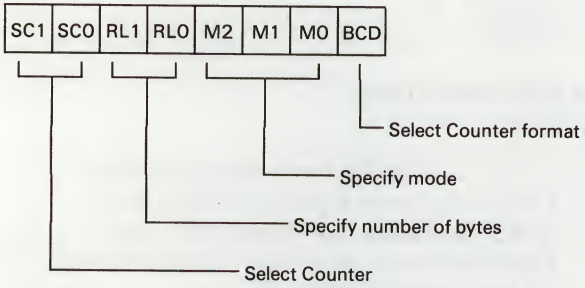
M2	M1	M0		
0	0	0	Mode 0	Interrupt on terminal count
0	0	1	Mode 1	Programmable one shot
x	1	0	Mode 2	Rate generator
x	1	1	Mode 3	Squarewave rate generator
1	0	0	Mode 4	Software triggered strobe
1	0	1	Mode 5	Hardware triggered strobe

Counter format

BCD	
0	Binary Counter
1	BCD Counter

Counter Initialisation

Initializing a selected counter requires the control byte to be assembled as detailed below and then written to I/O address OEH.



Loading the Counters

Once initialized, the counters can be loaded with the count state at any time following initialization. The only constraint is that when more than one byte is to be loaded, the bytes must be loaded into the counter address location in the order specified by the control byte.

Selecting the condition for loading one byte only into the counter, automatically sets the other byte to zero count state. The counter begins decrementing after the full count value has been loaded.

Reading the counters

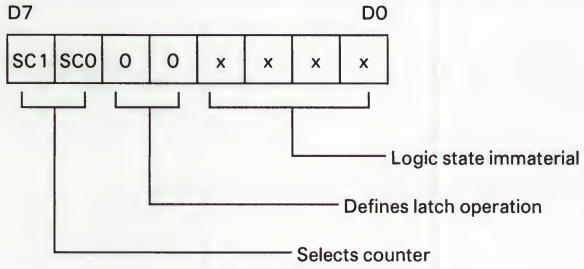
Two methods are available to read the count values reached by the counters, whilst the counters are counting down. Neither method has any affect on the operation of the counters.

The first method is achieved by reading the data stored at the selected counter address location using two consecutive read operations. The first read operation returns the least significant byte of the count state and the second, the most significant byte. This method can lead to problems as the count state is never stable.

The second method ensures that a stable count state reading is obtained by utilising the control byte to latch the count state into a storage register associated with each counter.

The count state is then obtained in the same manner as the first method. (By performing two consecutive read operations to access the stored count values at the counter address location).

To latch the count state into the storage register, the control byte is written into the address location of the control register (OEH), with the format as detailed below.



Programming Counter 0

For the output of Counter 0 (OUT 0) to be used as an interrupt request line to the Interrupt Controller, the counter must be set to operate in Mode 0.

When the counter is initialized, output (OUT 0) is set to logic low. After the counter is loaded with the count state, the counter begins to count down as illustrated in the timing diagram, Figure 2. (The BIOS programs the counter to produce a waveform with a 20 ms period).

The count state is decremented on the negative edge of each 2 MHz clock pulse. On reaching zero count state, the counter output is set to logic high, generating an interrupt request on the highest priority interrupt line connected to the PIC, IR0. This condition remains static until reset by the programmer, (either by re-initializing the counter or by loading a new value into the counter).

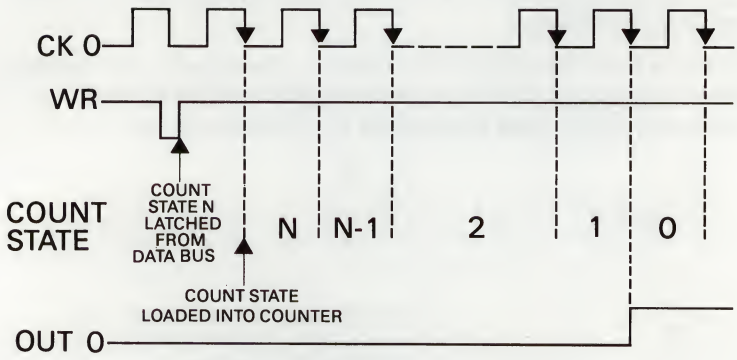


Figure 2. Mode 0 timing diagram.

The counter continues to decrement after the zero count state is reached, starting from the maximum count state of the counter, until it is reset. (2^{16} for a binary counter, 10^4 for a BCD counter). This feature allows the software to determine the exact time of the interrupt request by reading the Counter 0 count state and performing a simple calculation.

Programming Counter 1 and 2

Counter 1 and Counter 2 must be set to operate in Mode 3, to produce a squarewave output for use by the RS232C serial interface.

In Mode 3, a true squarewave is only produced by programming the counter with an even count state. Under this condition, the counter remains at logic high for one half of the count state, and at logic low for the second half. The counter is decreased by two on the negative edge of each 2 MHz clock pulse as illustrated in the timing diagram, Figure 3.

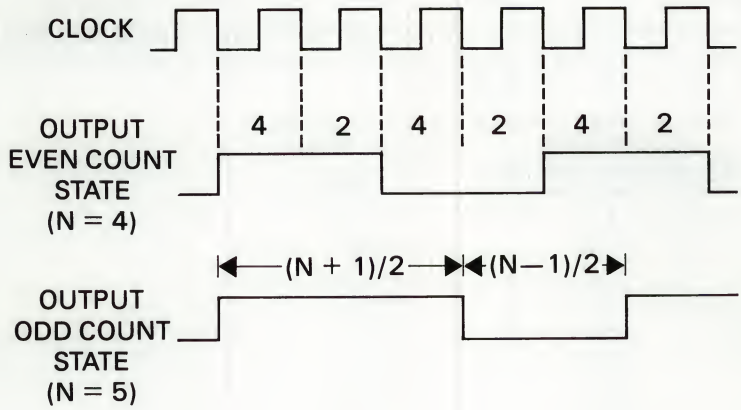


Figure 3.Mode 3 timing diagram

When the counter reaches zero count state, the original count value is automatically reloaded into the counter and the process repeated. This produces a squarewave output with a frequency equivalent to the input clock frequency (2 MHz) divided by the count state, and a 50% duty cycle.

When the counter is programmed to operate in Mode 3 using an odd count value, the frequency of the output is equivalent to 2 MHz divided by the count value as for an even count value, but the output waveform is asymmetrical. The asymmetry being more pronounced for low count values (high output frequencies).

If the value is odd, the counter output remains at logic high for (N+1)/2 clock pulses and remains at logic low for (N-1)/2 clock pulses as illustrated in the timing diagram Figure 3, where N represents the count value.

The count values for programming Counter 1 and Counter 2 to produce some of the commonly used baud rates, are detailed in tabular format below. These values only apply to asynchronous communications, where the outputs supplied from the counters are divided by 16 internally within the SIO prior to setting the baud rate.

Equivalent Baud Rate (Bauds)	Count Value N (Hex.)
50	09C4
75	0683
110	0470
134.5	03A1
150	0341
300	01A1
600	00D0
1200	0068
1800	0045
2400	0034
3600	0023
4800	001A
7200	0011
9600	000D
19200	0007

Sound Generation 2.10

Contents

Illustrations

Introduction

The Sound Generator is located on the Interface Board and consists of the elements of circuitry as illustrated on the block diagram below. The Sound Generator can be programmed to produce audio frequency tones, audio noise, or complex synthesized sounds in a frequency range from 500 Hz to 5 kHz. It is utilised by the BIOS to generate:

1. The keyboard keyclick.
2. A bell tone to act as a general warning signal by a number of applications.

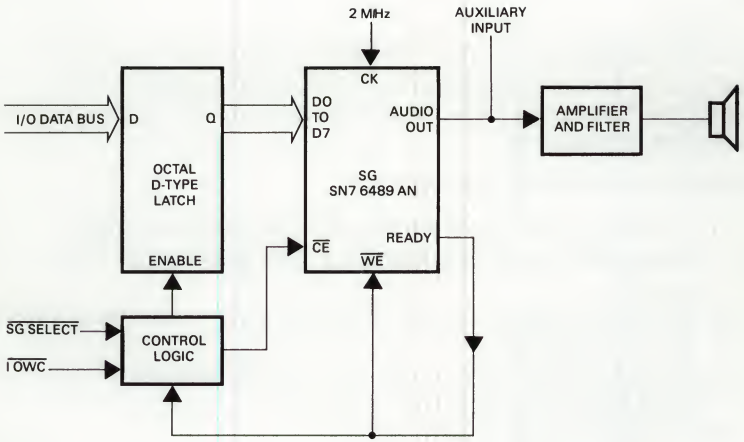


Figure 1. Sound Generator block diagram

SG Pin Definition

D0 to D7	Data bus connection
CK	2 MHz clock input
$\overline{CE}$	Chip Enable input
$\overline{WE}$	Write Enable input
READY	Ready status output
AUDIO OUT	Audio drive signal

Details

General

The Texas Instruments SN 76489 Programmable Sound Generator (SG) forms the basis of the sound generation circuitry. This is the same device as used on the Apricot *pc*/**xi** range of computers.

Internally the device consists of:

1. Three programmable tone generators, each with associated control registers. (Tone generators 1, 2, and 3).
2. A single programmable noise generator with associated control registers.
3. An 8-bit parallel interface for transferring data from the data bus to the control registers.
4. An audio output buffer stage.

Each generator consists of two cascaded sections (see diagram below); a section which produces the audio signal; and a programmable attenuator section which determines the audio signal output level. The attenuator section also provides the control for switching the generator off.

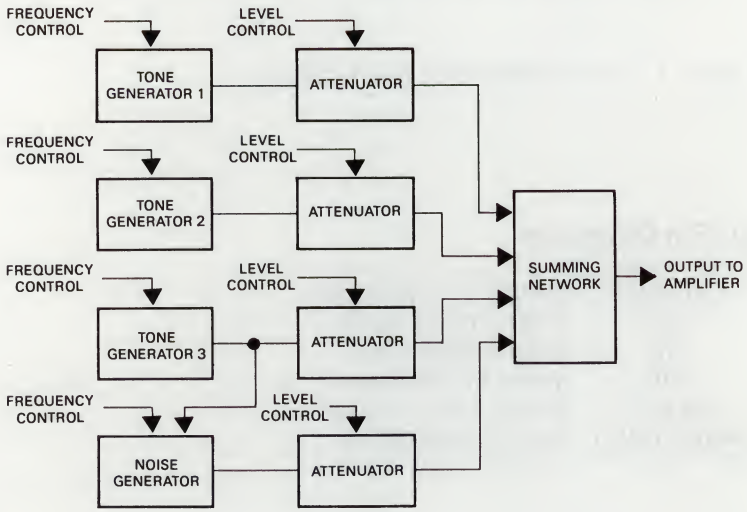


Figure 2. Sound Generator detail

The outputs of all four generators are automatically combined by a summing network which forms the audio output buffer of the device.

The system software views the Sound Generator (SG) as a single peripheral port, with an address location of 26H as defined by the SG select connection. Access to registers for controlling the audio output level and output frequency is determined by setting appropriate bits within the control bytes written to the port.

The SG Device has a relatively slow data loading time. Because of this, consecutive bytes of data written to the Sound Generator should normally be separated by a delay of at least 32 clock cycles of the SG device input clock frequency (16 μs, i.e. approximately 80 processor clock cycles). If the delay is not implemented, unpredictable tones and noises may result.

To match the slow data loading time of the Sound Generator to the speed of operation of the CPU without introducing processor wait states, the control data is not written directly to the SG device but latched through to the device inputs via an octal D-type latch. The SG then loads the control data stored at the outputs of the octal latch into the selected control register itself without any further action from the processor. This is done by the generator issuing a logic low state on the READY control output and utilising its associated control logic circuitry to extend the device write cycle time. The mechanism employed is detailed below.

The logic low state on the ready control output, indicating that the SG is transferring control data into an internal register, is automatically generated on receipt of a logic low state at the $\overline{CE}$ input. The effect produced by the logic low state on the READY control output is to extend the SG input write cycle time, by holding the write enable input in the active low state for approximately 32 periods of the 2MHz input clock frequency (supplied from the 6301 processor). The READY output returns to logic high on completion of the internal data transfer process.

An auxilliary input (two-pin molex) is provided into the sound amplifier circuitry which allows the internal speaker to be driven by another source. The maximum input level to the driver circuit is 50 mV peak-to-peak over the frequency range 500 Hz to 5 kHz.

Tone Generation

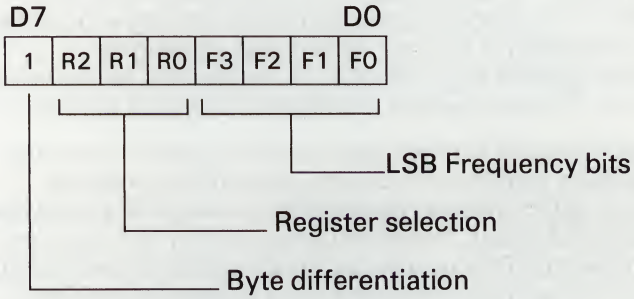
To generate a pure audio tone initially requires the tone generator to be programmed with three bytes of information, and the three other generators to be switched off. The first two bytes determine the frequency of the audio tone and the third byte determines the level of the produced audio tone.

Once programmed with the frequency data, the tone generator can be switched on and off, using the level control byte alone.

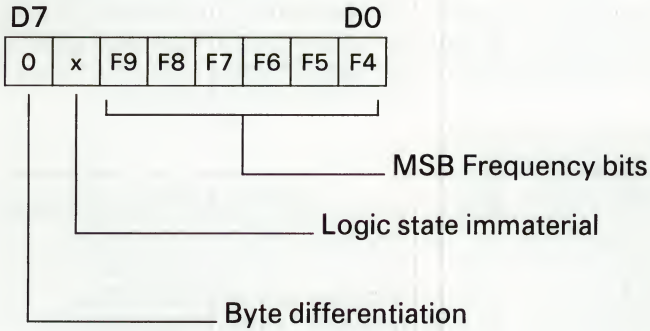
The frequency of the audio tone is set by a 10-bit word which forms part of the first two bytes. The remaining bits of the frequency bytes select the tone generator register and differentiate between the byte containing selection bits and the byte containing frequency data only. The format of the frequency bytes and the expression used to calculate the frequency of the tone is as follows.

$F = 10^6/16n$ where F is the tone frequency and n is a 10-bit binary number.

First Frequency Byte



Second Frequency Byte

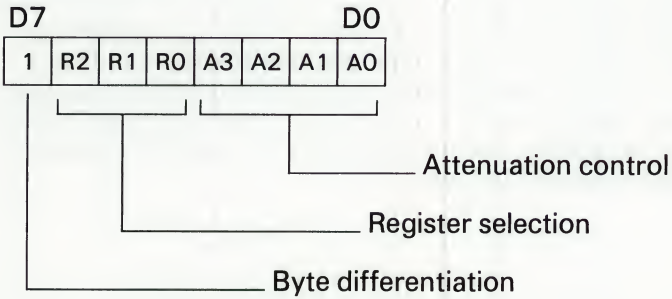


Register selection

R2	R1	R0	Register
0	0	0	Tone Generator 1 Frequency
0	0	1	Tone Generator 1 Attenuation
0	1	0	Tone Generator 2 Frequency
0	1	1	Tone Generator 2 Attenuation
1	0	0	Tone Generator 3 Frequency
1	0	1	Tone Generator 3 Attenuation
1	1	0	Noise Generator Control
1	1	1	Noise Generator Attenuation

The level of the audio tone is determined by a 4-bit word which forms part of the level control byte. Three of the remaining bits select the generator attenuator register (as detailed above), and the last remaining bit signifies that the byte contains generator register selection bits. The format of the level control byte is as follows.

Level Control Byte



The attenuation control bits allow 15 different levels of attenuation to be switched in from 0dB to 28dB in 2dB steps, and also allow the tone generator to be switched off. Each individual attenuation control bit introduces a different amount of attenuation as detailed below.

Attenuation Control

A3	A2	A1	A0	Attenuation
x	x	x	1	Introduces 2dB of attenuation
x	x	1	x	Introduces 4dB of attenuation
x	1	x	x	Introduces 8db of attenuation
1	x	x	x	Introduces 16dB of attenuation
1	1	1	1	Switches Generator OFF

Note: Tone generator 1 is normally assigned to produce the bell tone. The frequency of the bell tone (800Hz) is set up during initialisation and then activated and inhibited using the level control byte.

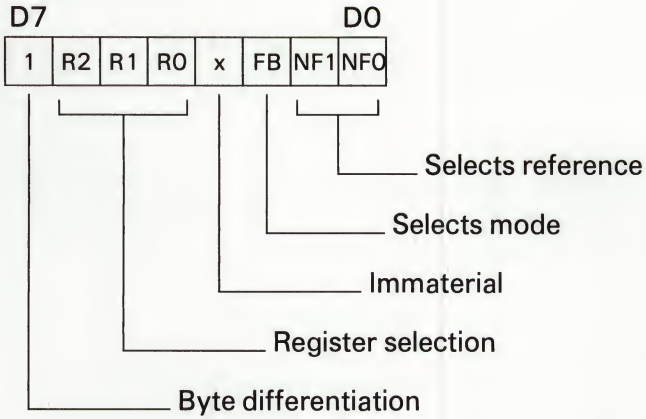
Noise Generation

The noise generator is able to function in either one of two modes. The first mode produces "white noise" and the second mode produces "periodic noise".

The noise source for the generator is based on a feedback shift register network with the shift rate determined by a reference frequency. The reference frequency can be either a preset frequency, or be determined by the audio signal frequency output from tone generator 3. A choice of three preset frequencies are available as shown in the Reference Selection table.

The choice of mode and reference frequency for the noise source is determined by a single noise control byte. Two bits of the noise control byte define the reference frequency and one bit defines the mode. Three of the remaining bits are used to select the noise generator control register (as defined by the register selection table in the tone generator section above), and the remaining bit signifies that the byte contains register selection bits. The format of the noise control byte is as follows.

Noise Control Byte



Reference selection

NF1	NF0	Reference
0	0	3.90 kHz
0	1	1.95 kHz
1	0	975 Hz.
1	1	Tone Generator 3

Mode selection

FB	Mode
0	"Periodic noise"
1	"White noise"

The level of the noise signal is determined by a level control byte which allows a choice of 15 different levels of noise signal to be set, and also enables the noise generator to be switched off. The format of the level control byte is identical to the format of the level control byte described in the tone generator section above.

Note: The noise generator is normally assigned to produce a keyclick, each time a key is pressed by updating both the noise generator register and the noise attenuator register.

Speech Interface 2.11

Contents

Introduction

This chapter describes the hardware used to implement the interface between the microphone and the voice driver circuitry. It is documented purely as interest value and for the sake of completeness.

It does not provide any great detail on the the 6301 processor and it's interface to the 8086 since the likelihood of anybody implementing a different speech driver using the same interface is extremely remote due to its inherent complexity.

Details

The Interface Board on the Portable contains an area of circuitry, which forms the interface between the user (via a microphone) and the speech driver software. It is based upon a analogue-to-digital (a-d) conversion stage, controlled by a 6301 processor. A block diagram of the speech interface system is illustrated on Figure 1.

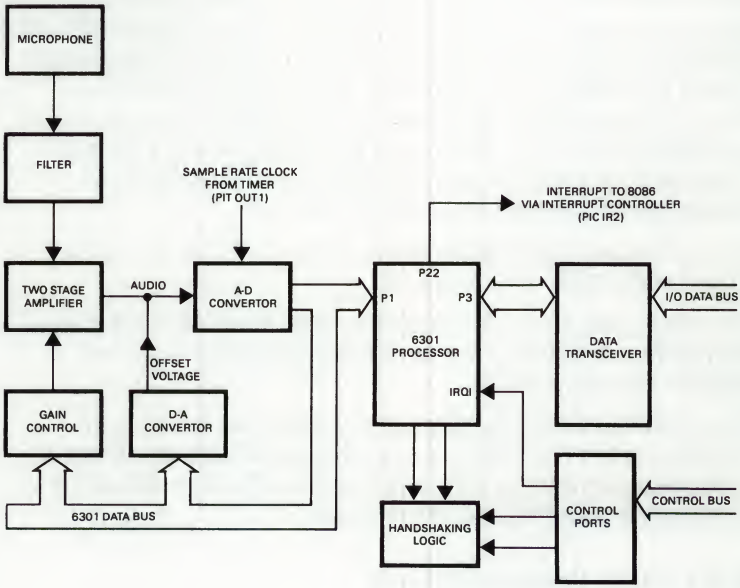


Figure 1. Speech Interface

The prime purpose of the interface is to convert the analogue speech input from the microphone into digital format for use by the speech driver software.

Circuitry

This consists of a multi-stage signal processing circuit. The first stage is a lowpass filter stage which acts as a pre-emphasis network, which increases the high frequency components of the audio frequency signal relative to the lower frequencies. (It is actually a passive lowpass filter with a cut-off frequency of 1600 Hz).

The second stage is a two-stage amplifier section, which also band limits the signal to within a frequency spectrum 340 Hz to 4 KHz.

The gain of the amplifier section is controlled by the 6301 processor. It does this by switching in different resistor values into the feedback network on the amplifiers. This allows various gain settings from 0 to 65 dB to be set in discrete steps.

The third stage in the circuit is the actual a-d converter which is based upon a sample and hold arrangement. The sample rate is controlled by a clock signal supplied from the on board Timer circuit. The same clock output from the timer can also be used for setting up the RS232C port for an internal split speed baud rate clock, but will affect the operation of the speech interface circuit.

The value programmed into the timer is set by the speech driver software.

An offset voltage is also supplied from the 6301 via a d-a converter for biasing the audio signal to set it into the required sample range.

The digitised audio samples are read by the 6301 and transferred to the 8086 processing environment using an elaborate communication mechanism. This involves a data transceiver and the use of interrupts.

6301/8086 Communications

When the 6301 has data to transfer to the 8086, it generates an interrupt which is supplied to the interrupt controller (PIC) on the CPU and Display Board (via interrupt request line - IR2).

The associated interrupt service routine clears the interrupt by writing to an I/O control port (12H). This is detected by the 6301 (by monitoring one of it's input ports) and acts as a "request for data" signal.

The 6301 sends a byte data to the transceiver which forms the interface between the 6301 and the 8086 data bus, and also sets a flag (LP23 - wired to bit 4 of the parallel port which is primarily used by the printer interface - I/O port 24H).

The 8086 polls the parallel port line D4 and reads the data from the transceiver (I/O control port 10H) on detecting an active signal on LP23. The final operation of the data byte transfer sequence is performed by the 8086. This is clearing LP23 by writing to another I/O control port (16H). This action also indicates to the 6301 that another byte of data can be transferred as necessary. The number of data bytes to be transferred is signified by the first bytes sent from the 6301.

When the 8086 wishes to send data to the 6301, it does so by generating a 6301 interrupt. This is achieved by simply writing to the I/O control port 14H and then after a delay writing to port 12H. The action of writing to 14H, produces an active low state on the 6301 Interrupt Request line IRQ1, via a D-type latch.

The 6301 interrupt service routine clears the interrupt and also sets the flag LP23 (wired to bit 4 of the parallel port as described above) to indicate it is ready to receive data.

The 8086 polls the parallel port line D4 and sends the data to the transceiver (I/O port 10H) on detecting an active signal on LP23. The 8086 then clears LP23 by writing to I/O control port 16H.

This signifies to the 6301 that data is available and enables it to read the data stored in the transceiver. This process continues with the 6301 signalling it is ready to receive data using the LP23 handshake mechanism depending on the number of bytes to be transferred. This is specified by the first bytes of data sent to the 6301.

Address Allocation

The following addresses located in the system I/O space are used for communicating with the 6301.

Address	Function	Access
10H	8-bit Data Port	Read/Write
12H	Interrupt Clear	Write only
14H	Generate 6301 Interrupt	Write only
16H	Clear LP23	Write only
36H	Generate 6301 NMI	Write only
24H	LP23 status (bit 4)	Read only

Ports 12H, 14H and 16H do not require data to be sent to generate the respective function. The 6301 NMI (non-maskable interrupt) is generated by writing FFH to the bit wide port 36H (only bit 0 is significant). It's effect is to re-initialise the speech interface circuitry.

Disk Drive 2.12

Contents

Illustrations

Introduction

This chapter provides information on the disk drive fitted internally within the Apricot Portable, the Sony OA-D32W. The drive is characterised by incorporating two read/write heads and utilising 80 track double-sided MicroFloppy disks. It is possible to use (format, read from/write to) 70 track MicroFloppy disks within the 80 track drive. (The drives are physically compatible with 70 track MicroFloppies and the BIOS software is configured to support this feature).

The disk drive is mounted directly onto the plastic assembly which forms the base of the Portable, and secured in position by four screws accessible from the underside of the Unit.

Connections from the disk drive controller, the Floppy Disk Controller on the Interface Board, are linked to the drive via a 26-way ribbon cable assembly. Power supply connections to the drive originate from the Power Supply Unit and are provided by a 4-wire cable assembly.

Note:

To avoid the possibility of damage occurring to the read/write heads of the drives within the computer during transit, packing disks should always be inserted into the drive slots prior to transportation. If packing disks are not installed prior to transportation, excessive vibration during transit may cause the heads to crash together resulting in unrepairable damage.

Do not:

1. *Switch the machine on with the packing disk in the drive slots.*
2. *Insert the packing disks into the slots whilst the power supplies are present.*

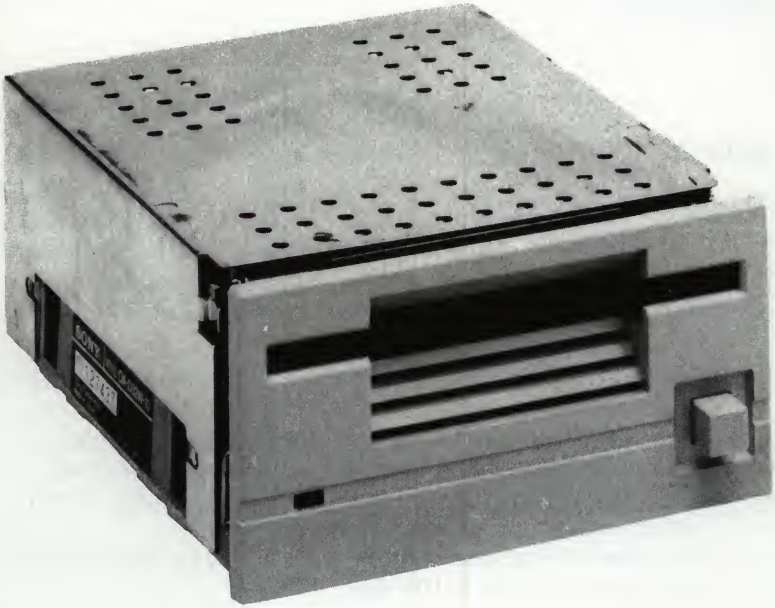


Figure 1. MicroFloppy Disk Drive

Details

General

The disk drive contains all the necessary electronics and mechanics for transferring MFM encoded serial data between the MicroFloppy disks and the Interface Board.

The electronics consist of:

1. The interface to the disk controller (housed on a single printed circuit control board located at the base of the drive).
2. A series of sensors for detecting various conditions within the drive (e.g. When the heads are positioned over the first track defined as Track 0 of the disk, when a disk is in the drive, etc.).
3. The read/write head tranducer circuitry for reading and writing data from/to the disk.

The mechanics consist of the disk drive mechanism, the disk loading/eject mechanism, and the mechanisms for positioning and engaging the read/write heads.

Interface Details

Connections between the disk drive and the Interface Board consist of four types; MFM encoded data signals, control input signals, status output signals and power supply lines. The latter is supplied via the four wire cable assembly; the remaining three types via the 26-way ribbon cable.

All signals supplied via the 26-way ribbon cable are driven by open collector 74 series logic gates, apart from the Index Pulse, which is driven by an open collector transistor.

The function of each connection is detailed in tabular format following the interface connector location diagram.

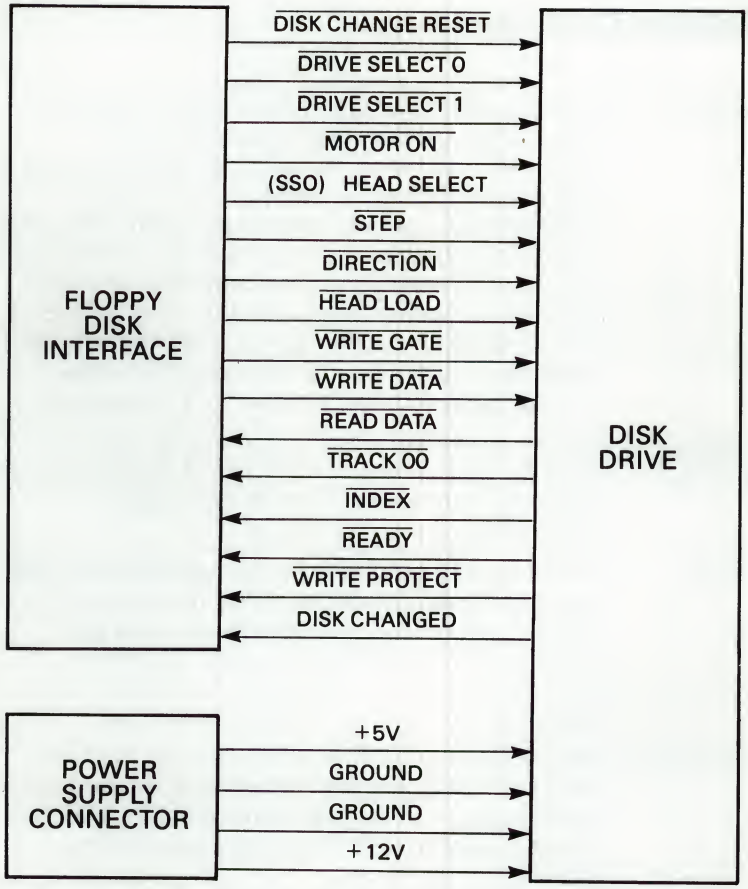


Figure 2. Interface block diagram

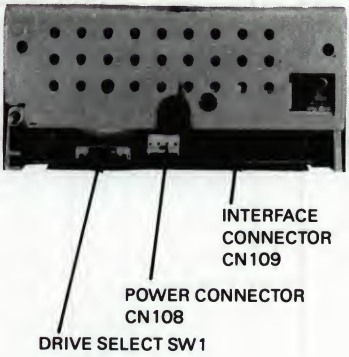


Figure 3. Connector Location

Interface Connections (Outputs)

Ready	Active state, logic low. When active, indicates that a disk is within the drive, the drive is selected, and the drive motor is rotating at the normal operational speed (i.e. the drive is available for a data transfer operation). The time taken for Ready to be set active after receiving the input to select the drive with the two other conditions already met, is of the order of 0.5 μs. The time taken for Ready to be set active after inserting a disk into the drive with the drive already selected, is the order of 1.6 seconds.
Write Protect	Active state, logic low. When active, indicates that the disk is write protected. If the drive is not selected, the output is set to logic high, regardless of the disk status.
Index	Index pulse which acts as the reference for the start of a track. Short duration negative going pulse (150 to 350 μs), generated once per revolution of the disk (i.e. every 100 ms at normal operational speed). If the drive is not selected, the output is held at logic high.
Track 00	Active state, logic low. When active, indicates that the heads are positioned over the first track on the disk. If the drive is not selected, the output is set to logic high regardless of the position of the head.
Read Data	MFM encoded serial data stream from the disk during disk reads. If the drive is not selected, the output is forced to logic high.
Disk Change	Active state, logic low. Output line which indicates when a disk has been changed. Generated by the action of ejecting a disk from the drive. Remains in the active state until a Disk Changed Reset pulse is supplied to the drive.

Interface Connections (Inputs)

$\overline{\text{Drive Selects}}$	Drive Select 0 and Drive Select 1. These outputs provide the capability for selecting one of the drives in a dual disk drive environment. The combination of states on these lines determine which drive is selected for operation. Normal decoded select status from the Interface Board is one Drive Select set low with the second select line set high. If Drive Select 0 is set low, the drive configured as drive 2 by a switch at the rear of the unit is selected for operation (see Drive Switch Settings for more details). Conversely, if Drive Select 1 is set to logic low, the drive configured as drive 3 is selected for operation.
$\overline{\text{Motor On}}$	Active state, logic low. Hardwired to 0V on the Interface Board. The drive is configured by a switch on the underside of the unit so that the drive motor is switched on only when a disk is within the drive.
$\overline{\text{Disk Change Reset}}$	A negative going pulse of 300 μs duration used by the programmer to reset the Disk Changed output line (see above).
$\overline{\text{Step}}$	A negative going pulse generated by the disk drive controller which moves the read/write heads, if the drive is selected. Each pulse causes the heads to be moved to an adjacent track location, in the direction specified by the direction input.
Direction	For each valid step pulse, the head moves in one track location towards track 79, if the direction control line is at logic low; and one track location towards track 0, if at logic high. If the head is already at either track 0 or track 79 and a step pulse is issued with the direction input set to move the head outside the normal track range, the head is held stationary.

Head Select	Control signal used to select the side of disk for the data transfer operation (Side Select output from the disk controller). A logic low sets the lower read/write head into the active state enabling data to be transferred to/from Side 0 of the disk (providing Head Load is active). Conversely, a logic high sets the upper read/write head active enabling data to be transferred to/from Side 1 of the disk (providing Head Load is active). The time taken for a read/write head to become ready for data transfers, following a change of state of the Head Select signal is 100 μs.
$\overline{\text{Head}}$ Load	Active state, logic low. When active, causes the read/write heads to make contact with the disk surfaces, providing the drive is selected. If the drive is deselected, whilst the head load signal is still active, the head remains loaded.
$\overline{\text{Write}}$ Gate	Active state, logic low. When active; enables the drive write control circuits to receive the write data from the disk controller; switches current through to the read/write head selected by the Head Select signal; and also enables the selected head's tunnel erase head. Set to the inactive high state during disk read and all head positioning operations.
Write Data	MFM encoded serial data. Changes the polarity of the current flowing through the selected read/write head on each negative-going transition, providing the following conditions are met: 1. The drive is selected. 2. The Write Gate input is active. 3. A write unprotected disk is inserted. 4. The drive motor is rotating at operational speed. 5. The heads have been loaded and are stationary.

Disk Drive Mechanism

The disk drive mechanism is a brushless direct drive motor, which employs a velocity servo control circuit to ensure that the disk rotates at a constant speed of 600 rpm. The drive is configured so that the motor rotates only when a disk is within the unit. Removal of the disk from the drive causes the motor to stop. The time taken for the motor to reach the normal operating speed following the insertion of a disk, is the order of 1.6 seconds.

The servo control circuit also generates the index pulse once per revolution of the disk.

Read/Write Heads

The two heads each consist of; a read/write element and a pair of tunnel erase heads, and are mounted opposite each other on the head guide arm. The tunnel erase section provides guard bands for adjacent tracks. Current is supplied to the read/write element of the selected head, on receipt of an active Write Gate signal from the disk controller which also activates the tunnel erase section.

Head Positioning Mechanism

The head positioning mechanism uses a stepping motor and a guide arm controlled by a needle screw to precisely position the read/write heads over the tracks on the disk. Control of the movement of the heads is supplied by the Step and Direction inputs from the disk controller. On application of the power supplies, the drive automatically generates control signals to position the heads over Track 0.

Head Load Mechanism

Head loading is controlled by the Head Load signal from the Interface Board. When the signal is active, one head makes physical contact with the lower surface of the disk; the second head makes physical contact with the upper surface. The Disk indicator on the front panel of the unit remains illuminated, as long as the head remains loaded. Head selection is determined by the Head Select input. This selects the side of the disk for a transfer operation by activating the read/write head tranducer circuits.

Sensors and Detectors

A series of photo-sensors and associated detector circuits are fitted in the drive. These generate status output signals to the disk controller, on detecting the conditions detailed below.

1. A disk is within the drive (Ready).
2. The disk is write protected (Write Protect).
3. The heads are positioned over Track 0 (Track 00).
4. The start of each track (Index Pulse).

Drive Switch Settings

Two switches are located on the drive, which are set according to the application of the drive within the system. One switch (SW S101) determines which of the two drive select input signals switch the drive to an operational condition. The second switch (SW S102) determines the method of switching on the disk drive motor.

The drive select switch is a 4-position switch located at the rear of the unit, as illustrated below.

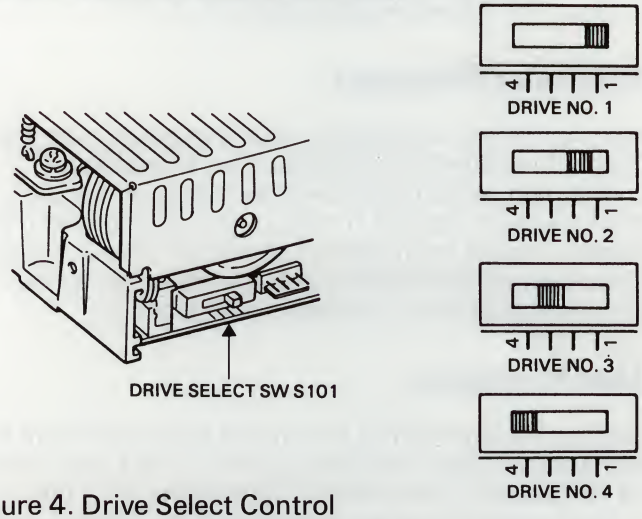


Figure 4. Drive Select Control

This switch (in theory) provides for four drives to be daisy-chained, with each one able to be uniquely selected by the two drive select input lines.

The decoding circuit of the Floppy Disk Interface circuitry on the Interface Board only allows two different combinations of states on the drive select input lines. These are drive select 0 at logic low, with drive select 1 at logic high; and drive select 0 at logic high, with drive select 1 at logic low. Due to this fixed decoding on the drive select inputs, the switch has to be set to the positions as detailed in the following paragraph.

In the standard single disk drive system, the switch on the internal drive is normally set to drive position 2. If an external second drive is daisy-chained with the internal disk drive, this is normally configured as Drive 3.

The motor control switch is located on the circuit board in the base of the unit, just behind the front panel (see illustration below). The switch must be set to position B, so the disk motor rotates only when a disk is within the drive. Setting the switch to position A, will cause the disk motor to rotate, regardless of whether a disk is within the unit or not.

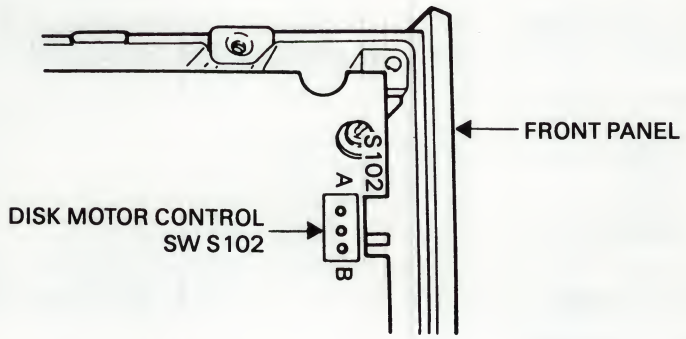


Figure 5. Drive Motor Control

Drive Specification

Media	3.5 inch 80 track double-sided MicroFloppy Disks. Soft-sectored with 9 sectors per track, 512 bytes per sector, producing a total formatted data capacity of 720 Kbytes per disk. Encoded with double density MFM data.
Data transfer rate	500 kbits/s.
Media life	More than 3×10^6 passes/track.
Track density	135 tracks per inch.
Track-to-track access time	12 ms.
Head load time	60 ms.
Head settling time	30 ms.
Head select time	100 μs. (Time taken for a read/write head to become operational, following a change of state on the Head Select input.)
Disk rotational speed	600 rpm (power-up time approx. 1.6 seconds after insertion of the disk).
Dimensions:	
Height	2.0 inches (51mm).
Width	4.0 inches (102mm).
Depth	5.1 inches (130mm).
Weight	1.5 lbs. (700g).
Current consumption	+12V supply: Typically 0.3A (max. 1.5A) +5V supply: Typically 0.48A (max. 0.8A)
Environmental conditions:	
1. Operating	
Temperature	40F to 115F (5C to 45C).
Humidity	20% to 80% relative humidity, with a wet bulb temperature of 85F (29C) and no condensation.
2. Storage	
Temperature	-40F to 140F (-40C to 60C)
Humidity	5% to 95% relative humidity,(no condensation).

Disks

General

Note: To ensure complete compatibility with the Apricot, only ACT approved MicroFloppy disks should be used with the disk drive. Using non-ACT disks, not manufactured to the same high standard, may result in intermittent read and write errors, rendering information stored on the disk totally unintelligible.

The Sony OA-D32W disk drive are designed to use 3.5 inch 80 track double-sided MicroFloppy disks. The disks are encased in a rigid plastic shell and feature an automatic shutter and a metal centering hub.

The shutter protects the disk media from contamination by dust, dirt or fingerprints, allowing the disk to be easily handled without affecting the integrity of stored data. When inserted into the drive, the shutter automatically slides open, allowing the read/write head of the drive access to the recording media. Removal from the drive automatically closes the shutter.

The metal centering hub ensures that the disk is accurately positioned on the disk drive motor spindle.

Disk Precautions

The same precautions apply to the MicroFloppy disks as any other magnetic recording media.

Do not:

1. Touch the disk surface.
2. Allow the disk to be placed in the proximity of other magnetic materials or other sources of magnetic fields.
3. Expose the disks to heat or direct sunlight.
4. Attempt to clean the disk surface.

Disk Insertion/Removal

Insert the disk into the drive shutter side first, with the metal centering hub facing downwards. The disk should slide into the drive with the minimum degree of force. Immediately the disk is inserted, the heads are momentarily loaded to seat the disk properly onto the drive spindle. This causes the front panel Disk indicator to be briefly illuminated. Improper insertion results in the disk not being accepted by the drive.

Remove the disk by pressing the disk eject button. The disk eject button should not be pressed, whilst the Disk indicator is illuminated.

Write Protecting

Write protecting the disk is achieved by sliding the tab (as illustrated below) to the lower position, creating a window in the lower left corner of the disk. To allow the disk to be written to once more, slide the tab back over the window.

Disk Format

Each of the 160 tracks on the disk is divided into sectors under software control (i.e. soft sectored), with the beginning of each track indicated by the index pulse generated by the drive motor assembly. The software track format chosen for the disks is a derivation of the IBM system 34 format for 8 inch disks. This uses double density MFM encoded data, with 9 sectors per track and 512 bytes per sector. The total storage capacity of the 80 track double-sided disk is thus 720 Kbytes of formatted data (160 x 9 x 512 bytes).

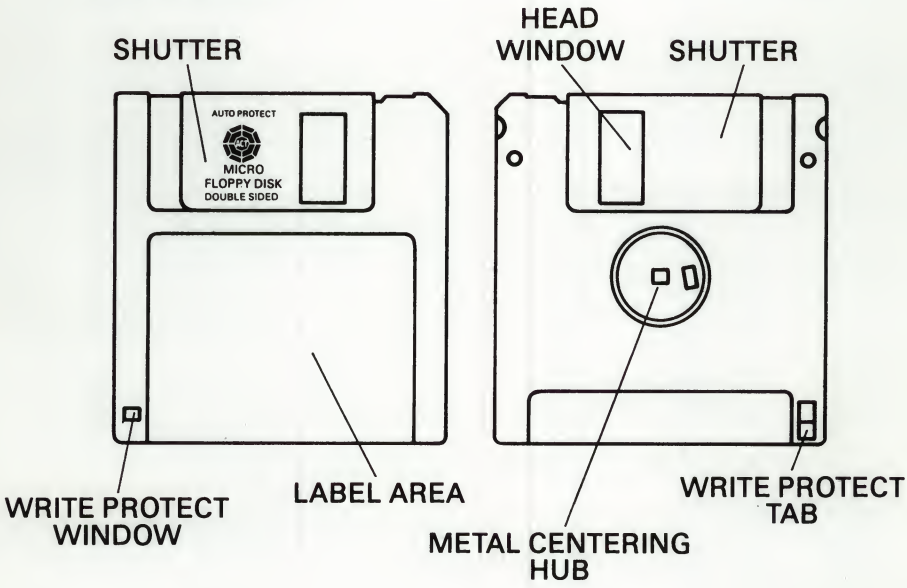


Figure 6. Microfloppy Disks

Keyboard 2.13

Contents

Illustrations

Introduction

The Keyboard for the Apricot Portable is a full function keyboard featuring; 92 keytops, four "special" keys, a time and date clock implemented in software, and an infra-red interface for transmission of keycodes and data to the Systems Unit.

It is an identical design to the Keyboard found on the Apricot F1 and is directly compatible and interchangeable. It employs the same circuitry, the same key layout, and uses the same keycode encoding scheme/data transmission rate as found on the F1 keyboard.

The maximum practical distance of operation is specified at 2 metres. The Keyboard will work at greater distances than this but the difficulties in viewing the screen so far away in any other mode apart from 40 column render this sort of usage of academic importance only.

A "light-pipe" is supplied with the Portable to connect the Keyboard and Systems Unit together in multiple machine environments where interference from other users may occur. One end of the pipe plugs into the recessed LED socket on the front edge of the Keyboard. The other end must be plugged into the right hand LED socket on the Systems Unit (as viewed from the front) to prevent interference from other infra-red sources.

The power supply for the Keyboard is provided by four AA cells. These are located behind a cover panel accessible from the underside of the Keyboard. The operational lifetime of the batteries is approximately 6 months under normal everyday usage.

Details

Mechanics

The mechanics of the Keyboard are of a comparatively simple design, consisting of a single circuit board, the main key switch array, the four fixed function keys and four AA batteries.

The batteries are located behind a removable cover panel in the base of the Unit. A coin or flat bladed tool is required to gain access to the battery compartment.

The Keyboard PCB is also fitted behind a removable cover panel. As user access is not necessary to this area, there is no easy coin-release slot.

The main key switch array is a single component which is clipped into the plastic moulding and cannot be removed once in position without damaging the array. It is linked to the circuit board by a multi-way ribbon cable connector.

The four fixed function keys are tracked directly onto the keyboard PCB.

The key tops of the key switch array are removable and can be easily repositioned for custom keyboard designs. Applying slight leverage underneath a key top releases it from its normal location, but care should be taken not to lose the springs.

Three infra-red LEDs are mounted on the Keyboard PCB and are located on the front edge of the Keyboard. The LEDs are spaced across the width of the Keyboard to provide a decent spread of infra-red signal. This removes any restriction on having to place the Keyboard directly in front of the Systems Unit, and thus allows the user the maximum degree of flexibility in siting the Keyboard.

Two of the LEDs are slightly proud of the front edge of the Keyboard, the third is recessed. The third LED is recessed to allow the light-pipe to be plugged in. Transmissions from the Keyboard are then supplied directly to Systems Unit down the fibre optic link.

The light-pipe does not switch off transmissions from the other two LEDs; these will continue to transmit infra-red when a key is pressed even though the light-pipe is in position. (Connecting the light-pipe into the right hand socket on the front of the Systems Unit switches off the infra-red receiver surrounded by the wide-angle lens, so that infra-red from other sources will not be detected - See Systems Unit chapter for more detail).

Two buttons are located on the sides of the Keyboard (one on each side). These release the spring-loaded feet which tilt the Keyboard for normal desk-top usage.

Circuitry

The circuitry on the Keyboard PCB consists of an NEC 7507 microprocessor, three infra-red LEDs and assorted components for interfacing the 7507 processor to the keyswitch array and LEDs. A circuit diagram of the Keyboard is provided in an appendix to this manual.

The 7507 is a CMOS 4-bit single chip microcomputer which has its own internal ROM and RAM. It also contains an internal timer, a vectored interrupt structure and features 32 I/O lines. The I/O lines are organised into eight 4 bit ports. The ports are identified by a prefix which is the port number followed by the port line number. e.g. P21 is port 2, line 1).

Keyboard Scanning

The 7507 uses the standard method of row/column scanning for detecting key closures in the keyswitch array and also the closure of the four fixed function keys. The four fixed function keys are mapped onto the bottom row of the keyswitch matrix (port P60).

The scanning method is as follows. The 7507 selects a row by setting the appropriate port output to logic low. It then sequentially scans through all the columns looking for a logic low input on the appropriate input ports which indicates a keyclosure.

The Shift and Control keys are not scanned in the same way and are treated as special keys. These are wired directly to input ports.

The keyboard processor is normally in a sleep mode and is awakened every 15.6 ms. The 15.6 ms timer routine performs two functions. It checks for key closures and also updates the software time/date clock.

If a key has been pressed, the keyboard remains awake and decodes the selected key, formats it, and then transmits it to the Systems Unit by pulsing the infra-red LEDs. If no other key has been pressed the keyboard re-enters sleep mode to conserve battery power.

When more than one key closure is detected in a single scan, the 7507 software checks the validity of the closures to prevent false key codes being issued. It does this by reducing the area to scan by searching a "closed" key switch map until it finds an unambiguous key closure.

Only keys which can be uniquely identified are encoded and transmitted.

The total time taken between detecting a key closure to the end of the keycode transmission is normally between 26 to 32 ms. At the end of the transmission, the processor rescans the key switch and then enters sleep mode if no further key is active.

The oscillator connected to the X1, X2 inputs of the processor sets the sleep mode cycle. The processor cycle time is set by the components connected to CL1/CL2.

Data Transmission Format

The majority of the consumption of power on the keyboard is taken by the three LEDs. In order to extend the battery life, these are normally switched off and only pulsed on for the transmission of data.

All keycode data is encoded into serial packet format, consisting of 32 bits, prior to transmission (see below for more details on the actual encoding format used). The 32 bit serial data stream is sent via the data output (P31) to drive the three LEDs.

The data is converted into a pulsed waveform by a monostable circuit prior to forming the drive signal for switching the LEDs on. The duration of the pulses and thus the time the LEDs are switched on, is kept relatively short (of the order of 15 μs) to minimise the amount of battery power consumed.

The monostable circuit translates the bit stream into the waveform format as shown in Figure 1. This is in effect a transmission packet consisting of an interleaved clock and data waveform. Encoding the data in this way allows the decoding circuits on the Systems Unit to easily compensate for variations in the data rate.

Data "1s" are signified by a pulse within the clock pulse time period and data "0s" signified by an empty time slot.

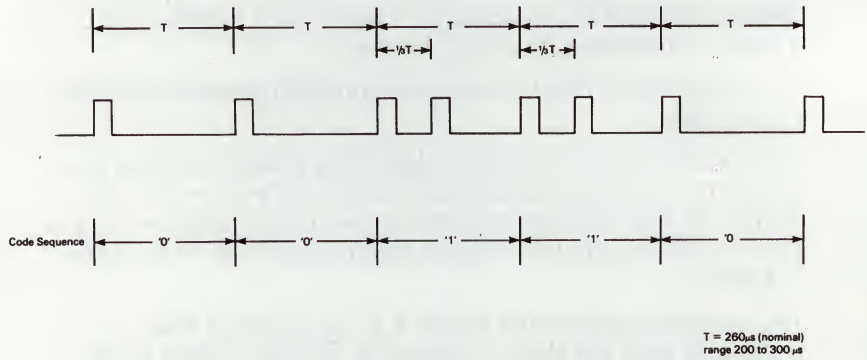


Figure 1.Keycode Transmission Format

Keycode Data Encoding

To maximise system reliabilty, and ensure that data transmitted from the Keyboard is not misinterpreted on the Systems Unit, the keycode data is encoded into a special format.

The Keyboard formats the valid key closures into a serial packet consisting of a four byte sequence. The format is the synchronous transmission mode Monosync and is operated at a fixed data rate of approximately 3.85 Kbits/sec. The first byte is the sync header. All the following bytes are the actual data. These are encoded using Hamming codes (see Figure 2).

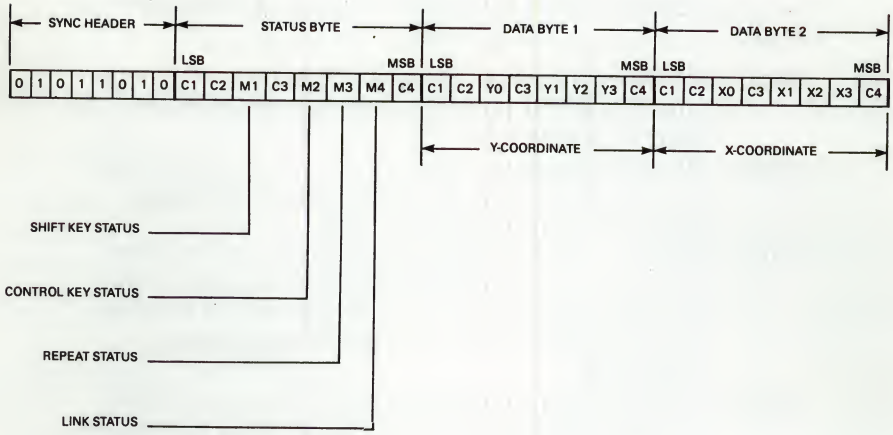


Figure 2. Synchronous Packet Format

This method enhances system reliability in two ways:

1. Using Monosync means that the receiver circuits on the Systems Unit (the Z80 SIO of the serial interface), have to first detect a valid data pattern, (the sync header byte) before it regards the data sent as being valid. This provides a high degree of protection from other infra-red sources, as they will not contain the sync header and will therefore be totally disregarded.
2. Using a Hamming format to encode the data enables the BIOS software in the Systems Unit to check the integrity of the data received from the Keyboard. It produces a highly reliable system for proving the validity of the Keyboard data, providing a measure of protection against a transmission which contains a valid sync byte, but invalid data (missing or corrupted data).

The four byte sequence consists of the Sync header byte (5AH), a status byte and two data bytes. The status byte and keycode data bytes are encoded with a Hamming format.

The status byte contains four bits (a nibble) of information as follows:

bit 0 Whether the key pressed is pressed in conjunction with the SHIFT key. This state is indicated when the Shift bit is set high.

bit 1 Whether the key pressed is pressed in conjunction with the CONTROL key. This state is indicated when the Control bit is set high.

bit 2 Whether the key code transmitted was the same key code as transmitted immediately prior to the current transmission. The Repeat Status bit is set high to indicate that a key is being held down (repeated).

bit 3 That the data transmitted to the Systems Unit is from the Keyboard. This bit is indicated by the Link Status bit and is always set low for keyboard transmission. (This bit allows the BIOS to differentiate between Keyboard and Mouse data which uses a similar format but sets the Link Status bit high).

The first keycode data byte contains a data nibble (Nibble 1) which is the Y-coordinate of the selected key in the keyswitch matrix. The second data byte contains a data nibble (Nibble 2) which is the X-coordinate of the selected key in the keyswitch matrix.

The X-Y coordinates for each key are illustrated on Figure 3. The coordinates are identified by the numbers in the top left-hand corner of each key. For example, the CAPS LOCK key is located at X-coordinate 2H, Y-coordinate BH (i.e. the keyswitch matrix position 2,11 in decimal format).

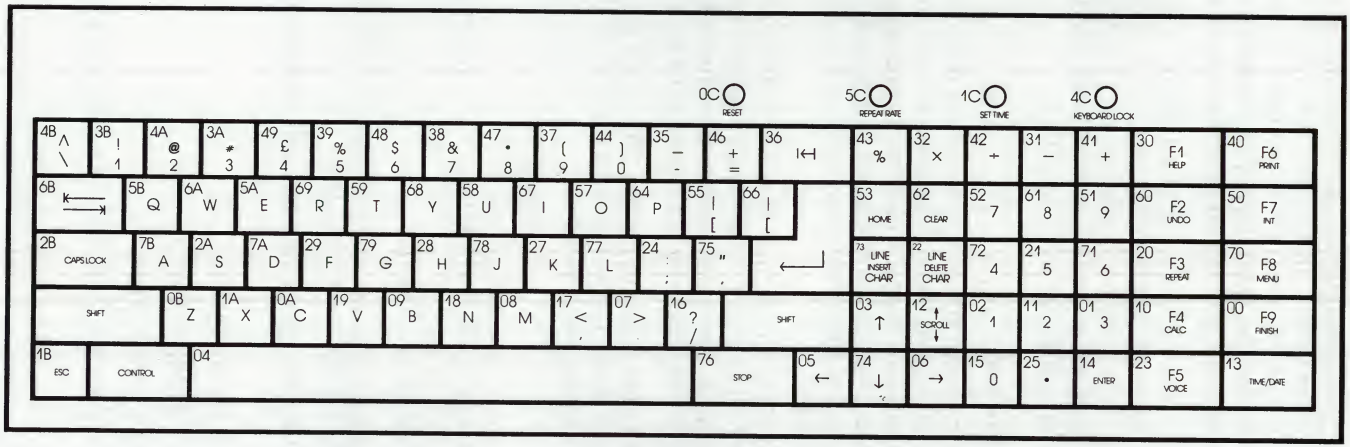


Figure 3. X-Y Coordinates

All keys apart from the RESET, REPEAT RATE, SET TIME, TIME/DATE (and SHIFT and CONTROL which are not in the matrix) are transmitted from the keyboard by encoding the X-Y position of the detected key with Hamming codes in the format as described above and perform no other function.

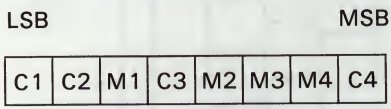
The handling of the RESET, REPEAT RATE, SET TIME and TIME/DATE keys are slighly different and are discussed under the section headed Special Keys. The SHIFT and CONTROL keys are encoded into the status byte only and are never transmitted as a X-Y keycode.

Hamming Format

The encoding of the status and keycode data nibbles uses a Hamming distance of four. (The definition of the Hamming distance equates to the fact that for all possible combinations of encoded bytes, each byte will have at least four bit positions different when compared with all other encoded bytes).

A Hamming distance of four allows the BIOS to detect any two-bit errors and also correct any single-bit errors in the transmitted data. Each data nibble is converted into a Hamming encoded byte by the addition of four check bits.

The Hamming encoded bytes are produced by generating the check bits and then combining them with the nibble of data in the format detailed below.



Hamming encoded Byte

C* corresponds to a check bit

M* is a bit within the data nibble. The position of the LSB is specified by M1, the MSB by M4.

The check bits are generated using the algorithms detailed below.

$$C1 = M1 \oplus M2 \oplus M3$$

$$C2 = M1 \oplus M3 \oplus M4$$

$$C3 = M2 \oplus M3 \oplus M4$$

$$\overline{C4} = C1 \oplus C2 \oplus M1 \oplus C3 \oplus M2 \oplus M3 \oplus M4$$

Since a nibble of data can only have one of 16 values (0H to FH), the data nibbles can be translated into the corresponding Hamming encoded bytes by using a simple look-up table. This is as follows:

Data Nibble	Encoded Byte
0H	80H
1H	07H
2H	19H
3H	9EH
4H	2AH
5H	ADH
6H	B3H
7H	34H
8H	CBH
9H	4CH
AH	52H
BH	D5H
CH	61H
DH	E6H
EH	F8H
FH	7FH

Special Keys

Some of the keys are not transmitted from the Keyboard in the same way as decribed above. These are the four keys RESET, REPEAT RATE, SET TIME and TIME/DATE.

Reset

When the 7507 microprocessor detects that the RESET key has been pressed it does not transmit a hamming encoded keycode. Instead, it sends sync characters (the byte value 5AH) at a set rate of one sync character every 15.6 ms for as long as the key is held down.

If the key is held down for approximately one second, the accumulative effect of receiving the multiple sync bytes on the Systems Unit generates a hardware reset. This mechanism is discussed in the System Detail chapter.

Repeat Rate

The Repeat Rate key is never transmitted to the Systems Unit. Instead it has the effect of acting as a toggle for varying the rate at which characters are sent from the Keyboard when a key is held down (i.e. the auto-repeat rate). It allows the rate of transmission to be either one of two values:

1. Eight characters per second which is the slow auto-repeat rate setting (one transmission every 125 ms).
2. Eighteen characters per second which is the fast auto-repeat rate setting (one transmission every 55 ms).

Set Time

The function of the SET TIME key is to allow the user to reset the time and date of the internal keyboard clock.

When the SET TIME key is pressed, the 7507 stops the internal clock and transmits the SET TIME keycode to the Systems Unit. The ROM BIOS on the Systems Unit decodes this key and activates the 25th line on the display to prompt the user to type in a new time and date. This action by the BIOS happens both before and after boot.

The user must then type in the time and date as a string of digits using the numeric keypad. Each digit is transferred to the Systems Unit in the normal four-byte Hamming encoded format. Any non-numeric key during the programming sequence causes the 7507 to restart the internal clock and transmit the non-numeric keycode. This informs the BIOS to terminate the operation.

The user must type in the sequence in the order detailed below with no spaces or other character between the digits (e.g. 1000011285 sets the clock for 10 am on the 1st December 1985).

HHMMDDMMYY

At the end of the 10 character sequence, the user must then press any key to set the internal clock to the specified time and date and also restart it. (Seconds are automatically set to zero by this action).

The updated time and date information is not used by the BIOS unless the information is re-transmitted by the Keyboard. This action is performed by the TIME/DATE key.

Time/Date

The function of the TIME/DATE key is primarily to transfer the time and date generated by the internal Keyboard clock to the ROM BIOS clock. It also performs another function, that of transmitting the version number of the Keyboard software.

When the key is pressed, the 7507 transmits a sequence of 16 data packets to the Systems Unit. Each packet is in the usual four byte synchronous format and are separated by 35 ms intervals. The first data packet is the TIME/DATE keycode packet. The following packets include the software version number and the time and date information.

The format of the sixteen data packet sequence is as follows:

Packet	Pre-Hamming Code Format*	Function
1	S031	Time/Date keycode
2	S41x	Revision number
3	S42x	Hours (tens)
4	S43x	Hours (units)
5	S44x	Minutes (tens)
6	S45x	Minutes units)
7	S46x	Seconds (tens)
8	S47x	Seconds (units)
9	S48x	Day (tens)
10	S49x	Day (units)
11	S4Ax	Month (tens)
12	S4Bx	Month (units)
13	S4Cx	Year (tens)
14	S4Dx	Year (units)
15	S4Ex	Not currently used
16	S4Fx	Not currently used

* The first value (S) is the Sync byte.

The second value is the Status nibble. The value of 4 in all the packets following the keycode corresponds to the Repeat Status bit set.

The third value in all packets apart from the Time/Date packet is a data packet identity nibble.

The fourth value (x) corresponds to the data nibble as described in the Function Column.